

NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et de génie informatique

INTERPOLATION TEMPORELLE DES IMAGES
AVEC ESTIMATION DE MOUVEMENT
RAFFINÉE BASÉE PIXEL
ET RÉDUCTION DE L'EFFET DE HALO

Thèse de Doctorat
Spécialité Génie électrique

Thi Thuy Ha TRAN

Jury Philippe MABILLEAU (directeur)
 Chon-Tam LE DINH (directeur)
 Roger Goulet (évaluateur externe au programme)
 Jean Rouat (rapporteur)
 André Zaccarin (évaluateur externe à l'Université)

Sherbrooke (Québec) Canada

Septembre 2010

TV-2115



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN 978-0-494-75065-0
Our file *Notre référence*
ISBN 978-0-494-75065-0

NOTICE

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

AVIS

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

■+■
Canada

*À ma mère, à mon père
et à Charlie!*

RÉSUMÉ

Dans le présent travail, après un résumé de l'état de l'art, une nouvelle interpolation temporelle des images avec réduction de halo est proposée. D'abord, pour la télévision de définition standard, une estimation de mouvement dont la résolution est le pixel, est suggérée. L'estimation se fait par l'appariement des blocs, et est suivie par un raffinement basé pixel en considérant des vecteurs de mouvement environnant. La réduction de halo se faisant à l'aide d'une fenêtre glissante de forme adaptative ne recourt pas à une détection explicite des régions d'occlusion. Ensuite, pour la télévision à haute définition, dans le but de réduire la complexité, l'estimation de mouvement de résolution pixel ainsi que la réduction de halo sont généralisées dans le contexte d'une décomposition hiérarchique. L'interpolation finale proposée est générique et est fonction à la fois de la position de l'image et de la fiabilité de l'estimation. Plusieurs post-traitements pour améliorer la qualité de l'image sont aussi suggérés. L'algorithme proposé intégré dans un ASIC selon la technologie de circuit intégré contemporain fonctionne en temps réel.

Mots-clés Convertisseur du taux d'images Vecteur de mouvement à la résolution du pixel Réduction du halo Fenêtre de forme adaptative selon l'intensité et le vecteur de mouvement Décomposition hiérarchique Traitement en temps réel

REMERCIEMENTS

Je tiens à exprimer ici ma profonde reconnaissance

- au Professeur Chon-Tam Le Dinh, mon directeur de recherche, pour toute l'aide apportée,
- au Professeur Philippe Mabilieu, mon directeur administratif de thèse, pour l'encouragement et les conseils éclairés depuis la transition de direction

J'adresse mes remerciements les plus chaleureux

- aux dirigeants de l'Université de Huê, Viet-Nam et de la Faculté des Sciences qui m'ont permis de poursuivre mon doctorat au Canada
- à l'Université de Sherbrooke et à la faculté de génie pour l'accueil qu'ils m'ont réservé

J'exprime toute ma reconnaissance

- à Sign-Pro Inc pour le financement de la première année d'étude
- à mes anciens collègues, Phuc-Tue Le Dinh et François Rossignol, à Silicon Optix Inc et à IDT Inc, où j'ai réalisé la plus grande partie de ma thèse, et auprès desquels j'ai toujours trouvé de l'aide, de la compréhension et des conseils éclairés
- à M Louie Lee et aux directeurs de ses compagnies pour leur accueil et leur support

J'adresse également un témoignage de sincère gratitude à tous mes amis au Viet-Nam et au Canada, dont les noms n'apparaissent pas ici, pour les encouragements nombreux tout au long de ce travail

Enfin, je suis profondément redevable à ma famille qui m'aime toujours comme je suis. Merci de leur patience, de leur soutien et de leur amour

TABLE DES MATIÈRES

| | | |
|----------|---------------------------------------------------------------------------------------------------------------|-----------|
| 1 | INTRODUCTION | 1 |
| 2 | ÉTAT DE L'ART | 3 |
| 2 1 | Recherche exhaustive et non exhaustive | 4 |
| 2 2 | Interpolation temporelle utilisant un codec vidéo existant | 5 |
| 2 2 1 | Projection du vecteur de mouvement le long du mouvement | 6 |
| 2 2 2 | Transposition latérale | 9 |
| 2 2 3 | Améliorations existantes | 11 |
| 2 3 | Interpolation temporelle utilisant l'estimation bidirectionnelle | 16 |
| 2 4 | Conclusion du chapitre et recherche proposée | 24 |
| 3 | INTERPOLATION TEMPORELLE PROPOSÉE POUR SDTV - ESTIMATION DE MOUVEMENT BASÉE PIXEL ET HALO RÉDUIT | 27 |
| 3 1 | Avant propos | 27 |
| 3 2 | Interpolation d'image avec l'estimation de mouvement basée pixel et réduction de halo | 29 |
| 3 2 1 | Introduction | 29 |
| 3 2 2 | Proposed frame rate conversion | 32 |
| 3 2 3 | Experimental results | 45 |
| 3 2 4 | Conclusion | 48 |
| 3 3 | Traitements supplémentaires | 50 |
| 3 3 1 | Réduction de l'effet de halo dans un arrière-plan périodique et quasi statique | 50 |
| 3 3 2 | Post-traitement pour scintillement | 53 |
| 3 4 | Résultat | 58 |
| 3 4 1 | Réduction de saccade et de flou | 58 |
| 3 4 2 | Réduction de l'effet de bloc | 60 |
| 3 4 3 | Réduction de l'effet de halo | 61 |
| 4 | APPROCHE HIÉRARCHIQUE POUR HDTV..... | 65 |
| 4 1 | État de l'art | 65 |
| 4 2 | Recherche proposée | 66 |
| 4 3 | Structure générale du système hiérarchique proposé | 67 |
| | Remarque sur la complexité et la réalisabilité du système | 70 |
| 4 4 | Raffinement hiérarchique basé pixel pour vecteur de mouvement | 75 |
| 4 5 | Interpolation d'image générique proposée | 79 |
| 4 5 1 | Révision des concepts pour la compensation de mouvement | 79 |
| 4 5 2 | Interpolation à compensation de mouvement générique proposée | 82 |
| 4 6 | Résultats | 86 |
| 4 6 1 | Halo et Artefact de bloc dans la structure hiérarchique | 87 |
| 4 6 2 | Résolution apportée pour les vecteurs de mouvement au niveau hiérarchique supérieur | 89 |
| 4 6 3 | Interpolation générique | 94 |
| 5 | CONCLUSION | 99 |
| 5 1 | Contributions apportées | 99 |
| 5 1 1 | Sélection des vecteurs de mouvement à la résolution du pixel | 99 |
| 5 1 2 | Réduction de halo HMVR | 100 |

| | | |
|----------------------------------------------------------------|-------------------------------------------------------------|------------|
| 5 1 3 | Raffinement hiérarchique basé pixel du vecteur de mouvement | 100 |
| 5 1 4 | Interpolation générique d'image | 101 |
| 5 2 | Extensions des recherches futures | 102 |
| ANNEXE | | 103 |
| ANNEXE A – PATENT APPLICATION US 2009/0161010 A1 | | 104 |
| ANNEXE B – RÉFÉRENCE [TRAN et al , 2010] | | 121 |
| ANNEXE C – INSTRUCTIONS CONCERNANT LE DVD DES RÉSULTATS | | 127 |
| LISTE DES RÉFÉRENCES | | 129 |

LISTE DES FIGURES

| | | |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 2 1 | Projection le long du mouvement | 6 |
| Figure 2 2 | Phénomène de « trou » (a) et de « chevauchement » (b) | 7 |
| Figure 2 3 | Résolution du problème de trou de [Jeon <i>et al</i> , 2003] | 8 |
| Figure 2 4 | Transposition latérale du vecteur de mouvement du bloc $B(u,n-1)$ | 10 |
| Figure 2 5 | Méthode d'estimation de mouvement de [Fujiwara et Taguchi, 2005] | 12 |
| Figure 2 6 | Compensation adaptative selon la classification de mouvement [Lee <i>et al</i> , 2002] | 14 |
| Figure 2 7 | Artefact de bloc visible dans les zones indiquées en rouge | 15 |
| Figure 2 8 | Méthode BDME et son ambiguïté éventuelle des vecteurs de mouvement | 16 |
| Figure 2 9 | Séquence <i>Flower Garden</i> | 17 |
| Figure 2 10 | (a) BDME classique, (b) BDME avec bloc additionnel | 19 |
| Figure 2 11 | Amélioration de la méthode BDME classique avec quatre images à l'entrée | 20 |
| Figure 2 12 | Approche quatre trames de [Sugiyama <i>et al</i> , 2005] | 21 |
| Figure 2 13 | Convertisseur FRC complet pour SDTV | 26 |
| Figure 2 14 | Convertisseur FRC complet pour HDTV (à multi -niveaux hiérarchiques) | 26 |
| Figure 3 1 | Proposed FRC | 32 |
| Figure 3 2 | Illustration of forward MV used in MVS calculation | 34 |
| Figure 3 3 | Paris Sequence (Copyright (C) 1994, 1996, PictureTel Corporation) | 37 |
| Figure 3 4 | Windows contains Foreground (black pixel) relatively moving to the left, Occluded Background (grey), Non-occluded Background (white) | 39 |
| Figure 3 5 | Decision Map (a, b) | 41 |
| Figure 3 6 | <i>Football</i> sequence – From left to right $\alpha = 1, 0.8, 0.6, 0.4, 0.2, 0$ | 45 |
| Figure 3 7 | Comparison of PSNR performance | 47 |
| Figure 3 8 | Cropped images – <i>Foreman</i> sequence – frame 57 | 48 |
| Figure 3 9 | <i>Bus</i> sequence | 49 |
| Figure 3 10 | Artefact de halo dans l'arrière-plan périodique | 50 |
| Figure 3 11 | Schéma de la réduction de l'effet de halo dans un arrière-plan périodique et quasi statique | 51 |
| Figure 3 12 | Réduction de l'effet de halo dans l'arrière-plan périodique quasi statique | 53 |
| Figure 3 13 | Schéma bloc du post-traitement pour scintillement | 53 |
| Figure 3 14 | Post-traitement pour scintillement de la séquence <i>Cheerleaders</i> | 56 |
| Figure 3 15 | Réduction de saccade de la séquence <i>Pendule</i> | 59 |
| Figure 3 16 | Réduction de l'effet de bloc sur l'arrière-plan de la séquence <i>Car Gate</i> | 60 |
| Figure 3 17 | Réduction de l'effet de bloc autour de la pendule en mouvement | 61 |
| Figure 3 18 | Réduction du halo avec la séquence <i>Walking Girl</i> | 62 |
| Figure 3 19 | Réduction de l'effet de halo en comparaisant avec la méthode de YUVsoft | 63 |
| Figure 4 1 | Schéma global du système HDTV | 68 |
| Figure 4 2 | Schéma détaillé du système HDTV | 73 |
| Figure 4 3 | Système proposé à trois niveaux hiérarchiques | 74 |
| Figure 4 4 | Premier ensemble des neuf vecteurs de mouvement candidats | 76 |
| Figure 4 5 | Raffinement hiérarchique basé pixel pour les vecteurs de mouvement | 78 |
| Figure 4 6 | Pondération linéaire et non-linéaire des vecteurs de mouvement | 83 |
| Figure 4 7 | Schéma-bloc de l'intégrateur proposé | 85 |

| | |
|------------------------------------------------------------------------------------------|----|
| Figure 4 8 Réduction de l'effet de halo et de bloc avec la séquence <i>Walking Girl</i> | 88 |
| Figure 4 9 Une trame originale de <i>NewYork City Spin</i> et la zone d'intérêt découpée | 90 |
| Figure 4 10 Images ré-interpolées et leur image d'erreur respective | 92 |
| Figure 4 11 Images d'erreur multipliée par 3 en pleine résolution et PSNR associés | 93 |
| Figure 4 12 Pondérations linéaire et non-linéaire avec la séquence <i>Big Buck Bunny</i> | 96 |
| Figure 4 13 Pondération linéaire et non-linéaire avec la séquence <i>Man in Park</i> | 97 |

LISTE DES TABLEAUX

| | | |
|-------------|---------------------------------------------------------------------|----|
| Tableau 2 1 | Résumé des techniques de FRC étudiées | 23 |
| Tableau 3 1 | PSNR Performance Comparisons among Four Frame Interpolation Methods | 46 |
| Tableau 3 2 | Séquences incluses dans le DVD joint avec la thèse | 64 |
| Tableau 4 1 | Évolution des PSNR- <i>NewYork City Spin</i> | 89 |
| Tableau 4 2 | Séquences HDTV incluses dans le DVD joint avec la thèse | 98 |

LEXIQUE

| Terme technique | Définition |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| ME backward | ME backward ou ME rétrograde est l'estimation de mouvement utilisant l'appariement de bloc de la trame présente à la trame précédente |
| ME forward | ME forward ou ME progressive est l'estimation de mouvement utilisant l'appariement de bloc de la trame précédente à la trame présente |
| Fallback | Technique de repli |

LISTE DES SYMBOLES

| Symbole | Définition |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| γ | Fiabilité estimée du vecteur backward dérivée d'une statistique locale basée contexte Le complément $(1-\gamma)$ est la fiabilité du vecteur forward |
| a_f, a_b | Paramètres de pondération a_f sert à pondérer l'erreur de l'estimation du vecteur de mouvement forward F (équation (3.3) de l'article) et a_b pour le vecteur de mouvement backward B |
| a_R | Rapport absolu des pixels (équation (3.10) de l'article) |
| B | Vecteur de mouvement backward ou rétrograde |
| BadErr | Mauvaise erreur |
| b_f, b_b | Coefficients de pénalité respectifs dans les erreurs pondérées forward et backward WEF et WEB (équation (3.3) de l'article) |
| b_R | Rapport relative des pixels (équation (3.10) de l'article) |
| d_F, d_B | Facteurs de pondération des vecteurs de mouvement forward et backward dans l'interpolation de l'image située à la position α (Section 4.5) |
| $dv, dv1, dv2$ | Vecteurs de mouvement ou de déplacement |
| F | Vecteur de mouvement forward or progressif |
| Fligmap | Carte de scintillement détecté |
| F_o, B_o | Vecteurs de mouvement sorties du bloc « Correction de vecteur de mouvement pour halo » au niveau de basse résolution |
| F_p, B_p | Vecteurs de mouvement sorties du bloc « Interpolation adaptive du champ des vecteurs de mouvement » |
| F_r, B_r | Vecteurs de mouvements sorties du bloc « Raffinement de l'estimation de mouvement basé pixel » |
| F_s, B_s | Vecteurs de mouvements sortis du bloc MVS |
| IMV | Champ de vecteurs de mouvement |
| $I_{n-\alpha}$ | Image interpolée à la position alpha α |
| I_{n-1} et I_n | I_{n-1} et I_n sont deux images respectives à l'instant $n-1$ du |

| | |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | passé et à n du présent |
| $I_{p\alpha}$ | Image interpolée préliminaire à la position alpha α |
| MADF, MADB | Moyenne des erreurs absolues forward/backward |
| mHiL | Carte de détection de l'effet de halo dans un arrière-plan périodique et quasi statique |
| MLEB($\mathbf{x}; \mathbf{p}$) | Moyenne de l'erreur absolue backward |
| MLEF($\mathbf{x}; \mathbf{p}$) | Moyenne de l'erreur absolue forward (équation (3 5) de l'article) |
| MWEF, MWEB | Minimum des erreurs pondérées locales forward, backward WEB (équation (3 4) de l'article) |
| N_D | Nombre de pixels d'intensité similaire mais de différent MV par rapport au pixel considéré |
| N_R | Nombre de pixels d'intensité et de vecteurs de mouvement similaires à ceux du pixel considéré |
| N_{RD} | Nombre de pixels d'intensité similaire à celle du pixel considéré |
| \mathbf{p} | Coordonnées d'un bloc avoisinant au bloc courant \mathbf{u} |
| PERD | Structure périodique détectée |
| PER_{hd} | Version répétée dans un bloc de $(U \times U)$ du signal PERD de basse résolution |
| \mathbf{R} | Région de référence de la fenêtre courante |
| S_{1F}, S_{1B} | Premiers ensembles des neuf vecteurs forward et backward aux distances multiples de U facteur de conversion en haute définition (équations (4 1) et (4 2)) |
| S_{2F}, S_{2B} | Deuxièmes ensembles des huit vecteurs de mouvements forward et backward proches du vecteur de mouvement en cours (équations (4 3) et (4 4)) |
| $SADF_{\Delta}/ SADB_{\Delta}$ | Sommes normalisées des différences absolues dans la fenêtre 5×5 autour du pixel à haute définition considéré |
| SADF, SADB | Sommes normalisées des différences absolues forward et backward |
| SF, SB | Versions répétées respectives de moyenne des différences absolues forward (MADF) et de moyenne des différences absolues backward (MADB) à la résolution de pixel |
| SZ | Zone de recherche (<i>Search zone</i>) |

LISTE DES SYMBOLES

| | |
|--------------|--------------------------------------------------------------------------------|
| SZ_n | Zone de recherche à la trame I_n |
| SZ_{n-1} | Zone de recherche à la trame I_{n-1} |
| \mathbf{u} | Coordonnées du bloc courant |
| \mathbf{W} | Bloc d'image de dimension $W \times W$ |
| WEF, WEB | Erreurs pondérées locales forward, backward WEB (équation (3) de l'article) |
| \mathbf{x} | Coordonnées du pixel courant en colonne et ligne |
| α | Position de l'image interpolée |
| α_2 | Une position fictive pouvant être non linéairement générée |
| λ | Facteur de proportionnalité |

LISTE DES ACRONYMES

| Acronyme | Définition |
|---------------|-------------------------------------------------------------------------------------------------------------|
| ASIC | Circuit intégré spécifique pour application (<i>Application Specific Integrated Circuit</i>) |
| BDME | Estimation de mouvement bidirectionnelle basée bloc (<i>Block-based Bi-directional Motion Estimation</i>) |
| BMA | Algorithme d'appariement de bloc (<i>Block Matching Algorithm</i>) |
| BME | Estimation de mouvement basée bloc (<i>Block-based Motion Estimation</i>) |
| C/U | Recouvrement ou Découvrement (<i>Covered or Uncovered</i>) |
| CIF | Format intermédiaire commun (<i>Common Intermediate Format</i>) |
| Codec | Codeur et décodeur |
| DM | Carte de décision (<i>Decision Map</i>) |
| FFT | Transformée de Fourier rapide (<i>Fast Fourier Transform</i>) |
| Filtre A&R | Filtre binaire « Add and Remove » |
| FRC | Convertisseur du taux d'image (<i>Frame Rate Converter</i>) |
| FRUC | Convertisseur du taux élevé d'image (<i>Frame Rate Up-Converter</i>) |
| FS | Recherche exhaustive (<i>Full Search</i>) |
| HBMA | Algorithme hiérarchique d'appariement de bloc |
| HD | Haute Définition (<i>High Definition</i>) |
| HDTV | Télévision à haute résolution (<i>High-definition television</i>) |
| HMVR | Raffinement des vecteurs de mouvement pour halo (<i>Halo consideration Motion Vector Refinement</i>) |
| IC technology | Technologie des circuits intégrés |
| LCD | Affichage à cristaux liquides ALC (<i>Liquid Cristal Displays</i>) |
| LUT | Table (<i>Look Up Table</i>) |
| MAD | Moyenne des différences absolues (<i>Mean Absolute Difference</i>) |
| MC | Compensation de mouvement (<i>Motion Compensation</i>) |
| MCI | Interpolation avec mouvement compensé (<i>Motion-Compensated Interpolation</i>) |
| ME | Estimation de mouvement (<i>Motion Estimation</i>) |

| | |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| MPEG1, MPEG2 | Standard 1 de <i>Moving Picture Expert Group</i> , Standard 2 de MPEG |
| MV | Vecteur de mouvement (<i>Motion Vector</i>) |
| MV backward | Vecteur de mouvement rétrograde |
| MV forward | Vecteur de mouvement progressif |
| MVF | Filtrage des vecteurs de mouvements |
| MVPF | Post filtrage de vecteur de mouvement (<i>Motion Vector Post Filtering</i>) |
| MVs | Vecteurs de mouvement |
| MVS | Sélection de vecteur de mouvement (<i>Motion Vector Selection</i>) |
| MWE | Erreur locale pondérée minimum (<i>Minimum Weighted local Error</i>) |
| OBMC | Compensation de mouvement utilisant les blocs avec recouvrement (<i>Overlapped block Motion Compensation</i>) |
| PSNR | Rapport du signal crête sur bruit (<i>Peak Signal to Noise Ratio</i>) |
| RAVPCD | <i>Recent Advances in Video Processing for Consumer Displays</i> |
| SAD | Somme des différences absolues (<i>Sum Absolute Difference</i>) |
| SAD _{nor} | Somme normalisée des différences absolues |
| SAW | Fenêtre de forme adaptative (<i>Shape-Adaptive Window</i>) |
| SDTV | Télévision à la résolution standard (<i>Standard-definition television</i>) |
| SEA | Algorithme d'estimation de mouvement utilisant les éliminations successives |
| SIF | Format intermédiaire standard (<i>Standard Intermediate Format</i>) |
| SSIM | <i>Structural SIMilarity</i> (http //www ece uwaterloo.ca/~z70wang/research/ssim/) |
| TV | Télévision |

1 INTRODUCTION

L'interpolation temporelle consiste à créer une ou plusieurs images successives entre deux images existantes consécutives. L'interpolation temporelle est donc une technique très en demande pour plusieurs applications.

La conversion des formats standards vidéo de 60 à 50 Hz et vice versa est un exemple classique pour la vidéo. La conversion des très anciens films de 16 à 24 Hz dans certaines maisons de postproduction est aussi un exemple qui est, par contre, moins connu. La création des séquences aux mouvements ralentis, la restauration des segments de film abîmés dans les studios sont les autres exemples qui peuvent s'effectuer en temps différé. L'interpolation temporelle est aussi utile dans un système de transmission des séquences de vidéo à bas débit où la diminution des images et la ré-interpolation subséquente sont requises pour sauver la largeur de bande de transmission et/ou pour maintenir la qualité d'image d'un codec. Récemment, dans les LCD (*Liquid Cristal Displays*), l'interpolation temporelle avec compensation de mouvement est requise pour réduire les mouvements saccadés et les artefacts associés au faible temps de réponse des LCD.

Dans la littérature, il y a de nombreuses techniques pour la conversion du taux de trames ou d'images (*Frame Rate Converter FRC*). En général, elle peut-être réalisée par un simple filtrage temporel, par une répétition des trames [De Haan, 2000] ou par une interpolation à compensation de mouvement. Le filtrage temporel peut causer un dédoublement d'images qu'on appelle parfois l'effet fantôme ou « *ghosting* » dans le jargon technique, lorsque le mouvement entre deux images consécutives est assez grand. La répétition d'images crée par contre les mouvements saccadés. L'interpolation à compensation de mouvement (MC) représente un potentiel pour lisser les mouvements entre les images.

Cependant, l'interpolation à compensation de mouvement rencontre souvent deux grandes difficultés techniques : la précision ou la résolution spatiale des mouvements de vecteurs (MV) estimés et l'effet des zones d'occlusion. Contrairement à la compensation des mouvements (MC) dans un codec dont les erreurs de prédiction peuvent être recorrectées par le

codage différentiel, l'interpolation à compensation de mouvement requiert une certaine précision de l'estimation du mouvement surtout dans les zones de recouvrement ou de découvrément, afin de créer, autant que possible, une image naturelle sans artefact visuel

Le présent travail de recherches, s'adressant aux problèmes mentionnés, a pour triple but de fournir a) une estimation de mouvement (ME) à la résolution du pixel tout en évitant des calculs itératifs, b) une solution pour raffiner les vecteurs de mouvement dans l'image interpolée y compris les zones d'occlusion, c) une extension hiérarchique d'estimation de mouvement (ME) pour la télévision à haute définition (HDTV) ou encore à ultra définition tout en gardant la possibilité du raffinement des vecteurs de mouvement Il va de soi que les solutions proposées respectent l'objectif d'un traitement en temps réel

La présente thèse sera organisée comme suit le prochain chapitre sera consacré à une étude de l'état de l'art, le chapitre 3 sera pour l'estimation de mouvement basée pixel proposée ainsi que le raffinement des vecteurs de mouvement dans les zones de recouvrement/découvrement, le chapitre 4 sera pour une extension hiérarchique de l'estimation et de la compensation de mouvement et finalement, la conclusion pour les contributions apportées et les extensions futures de la présente étude

De plus, la thèse sera rédigée sous le format combiné comprenant une publication et au moins un chapitre rédigé de manière conventionnelle Le chapitre 3 sera constitué alors d'un avant-propos, de l'article complet intitulé « Frame Rate Converter with Pixel-based Motion Vectors Selection and Halo Reduction using Preliminary Interpolation » et accepté pour publication comme article régulier dans *IEEE Journal of Selected Topics in Signal Processing, IEEE JSTSP, Special Issue on Recent Advances in Video Processing for Consumer Displays RAVPCD April 2011*, et finalement, des post-traitements supplémentaires pour compléter l'interpolation d'images TV de la résolution standard

Étant donné que la présente étude se réalise pratiquement dans un milieu industriel, plusieurs brevets (en annexe A et en annexe B) ont été déposés par l'auteure pour fin de protection des propriétés intellectuelles Le chapitre 4, rédigé d'une façon plus conventionnelle, est inspiré en partie des brevets en question

2 ÉTAT DE L'ART

L'interpolation des trames avec compensation de mouvement comprend deux principales parties : l'estimation des vecteurs de mouvement (*Motion Estimation*, ME) suivie par l'interpolation d'image avec la compensation de mouvement (*Motion Compensation*, MC). Comme la première partie constitue la base de l'interpolation d'images, les recherches sont alors plus nombreuses dans l'estimation de mouvement que dans la compensation de mouvement.

L'algorithme par appariement de blocs (*Block Matching Algorithm*, BMA) est généralement choisi pour l'estimation de mouvement dans la plupart des approches, et ce, grâce à sa simplicité de concept et de son implémentation matérielle ou logicielle. Il y a peu d'approches pratiques d'estimation de mouvement autre que l'algorithme par appariement de blocs pour l'application de la conversion du taux d'image. Le flux optique est employé dans [Krishnamurthy et Woods, 1999], la corrélation de phase est suggérée dans [Burl *et al*, 2000], tandis que [Choi *et al*, 2006] emploie une estimation de mouvement par maillage (*mesh based*) en combinant avec l'algorithme par appariement de blocs (BMA). Cependant, ces dernières méthodes requièrent une grande complexité pour l'interpolation d'image. De plus, [Blanchfield *et al*, 2006] et [Wang et Vincent, 2003] développent une estimation de mouvement basée sur l'objet. Cependant, cette technique implique la segmentation et l'étiquetage d'objets qui requièrent une mémoire de plusieurs images et une bande passante supplémentaire. En outre, la correction de vecteur de mouvement proposée exige de multiples itérations basées sur l'image en entier qui ne conviennent pas nécessairement pour le traitement en temps réel.

À partir de cette section, la discussion sera réservée uniquement au BMA comme technique de l'estimation de mouvement. Dans un BMA, l'image est divisée soit en blocs disjoints, soit en blocs avec recouvrement. Les vecteurs de mouvement (*Motion Vectors*, MVs) sont associés à chaque bloc disjoint ou non recouvert. Tous les pixels dans chaque bloc ont un seul même vecteur de mouvement. L'algorithme BMA fonctionne de la manière suivante. Étant donné un bloc dans une image appelée image d'origine (*anchor frame*), l'appariement consiste à

rechercher dans une autre image appelée image de référence (*reference or target frame*) un bloc de même dimension qui ressemble le plus au bloc original. La ressemblance peut être mesurée par une erreur de prédiction de mouvement. La somme (souvent normalisée par la dimension du bloc) des différences absolues (SAD) est la métrique couramment utilisée. La recherche s'effectue dans une zone de la deuxième image de référence. Si la recherche se fait en tout point dans la zone, elle sera appelée recherche exhaustive, sinon on l'appelle recherche non exhaustive. La recherche exhaustive permet de garantir une erreur de prédiction de mouvement minimale au moins dans la zone de recherche donnée. La recherche non exhaustive est souvent sous-optimale. Son avantage réside dans la réduction de la complexité du calcul.

Dans le présent chapitre, après un bref résumé concernant la recherche exhaustive et non exhaustive, l'étude de l'état de l'art sera consacrée aux diverses techniques d'estimation de mouvement pour l'interpolation temporelle d'image.

2.1 Recherche exhaustive et non exhaustive

Dans une interpolation temporelle d'image, la recherche exhaustive exige beaucoup d'opérations et de calculs, surtout pour la haute définition. Habituellement, les processeurs d'image n'ont pas suffisamment de bande passante ou de capacité pour mettre pleinement en œuvre cette opération. Afin de réduire la complexité, plusieurs techniques différentes d'estimation de mouvement ont été proposées. Liu et Zaccarin [Liu et Zaccarin, 1993] proposent une méthode efficace basée sur les pixels sous-échantillonnés au niveau de bloc au lieu d'un sous-ensemble de points de recherche. Les auteurs minimisent les effets du repliement de sous-échantillonnage des pixels par un filtre passe-bas. Li et Salari [Li et Salari, 1995] ouvrent une nouvelle classe d'algorithmes de recherche exhaustive rapide en utilisant les inégalités mathématiques basées sur la norme de la somme pour éliminer les blocs invalides de référence. La méthode proposée s'appelle l'algorithme des éliminations successives SEA (*Successive Elimination Algorithm*).

Dans le présent état de l'art des circuits intégrés, la méthode SEA et/ou ses similaires sont exécutables en temps réel pour la télévision à la définition standard SDTV, lorsque la zone de

recherche est de taille « raisonnable » Pour cette raison, sauf indication contraire, l'estimation de mouvement dans le présent chapitre est considérée effectivement exhaustive

Les algorithmes d'estimation de mouvement exhaustifs ou non procurent à l'image d'origine les vecteurs de mouvements basés-bloc Par contre, dans l'application de FRC, il est primordial de déterminer les vecteurs de mouvement dans l'image à interpoler qui n'existe pas encore En général, il y a deux grandes façons possibles de le faire

- a Obtenir le champ des vecteurs de mouvement de deux trames d'entrées par une estimation de mouvement forward et/ou backward Il est bon de noter que le champ des vecteurs de mouvement backward ou rétrograde est peut être déjà fourni si on peut avoir accès à un codeur de vidéo Ensuite, on projette le champ des vecteurs de mouvement de la trame d'entrée à la trame d'interpolation le long de l'orientation du vecteur de mouvement La projection du champ de des vecteurs de mouvement le long du mouvement produit les problèmes de « trou » et de « chevauchement » à examiner un peu plus tard Pour éviter ces problèmes incommodes, la projection ou encore la transposition latérale connue dans les décodeurs d'image est utilisée
- b Obtenir le champ des vecteurs de mouvement de la trame d'interpolation par une estimation de mouvement bidirectionnelle basée bloc (*Block-based Bi-directional Motion Estimation*, BDME)

2.2 Interpolation temporelle utilisant un codec vidéo existant

Lorsqu'un décodeur vidéo est disponible, pour en réduire la complexité, on peut utiliser les vecteurs de mouvement déjà estimés et transmis Comme ces vecteurs de mouvement sont dédiés à une image d'entrée, il s'agit alors de déterminer le champ des vecteurs de mouvement de l'image à interpoler à partir des vecteurs de mouvement fournis par le codec Pour faciliter la compréhension, la présente section peut se diviser en trois parties dont les deux premières représentent les deux techniques fondamentales Il s'agit de la projection le long de la trajectoire du mouvement, de la transposition dite « latérale » et des améliorations existantes

2.2.1 Projection du vecteur de mouvement le long du mouvement

Pour convertir les vecteurs de mouvement dans l'image existante I_n à ceux dans l'image interpolée, la première idée consiste à projeter le vecteur de mouvement le long de sa trajectoire. Soient I_{n-1} et I_n qui sont deux images respectives à l'instant $n-1$ du passé et à n du présent. Soit \mathbf{dv} le déplacement d'un bloc $B(\mathbf{u}, n-1)$ de coordonnées \mathbf{u} de l'image I_{n-1} à l'image I_n . Soit \mathbf{IMV} le champ de mouvements de I_{n-1} à I_n (ME forward). $\mathbf{IMV}(\mathbf{u}, n-1)$ correspond donc au vecteur de mouvement du bloc aux coordonnées \mathbf{u} dans le plan $n-1$.

On suppose que l'intervalle de temps entre deux trames consécutives est assez court pour considérer que le déplacement d'objet est une simple translation. La figure 2.1 illustre la façon de projeter le vecteur de mouvement le long de la trajectoire du mouvement. On peut l'appeler « projection le long du mouvement » pour être bref. Sur le plan à interpoler $n-1/2$, par exemple, le vecteur de mouvement du bloc $B(\mathbf{u}+\mathbf{dv}/2, n-1/2)$ sera attribué à $\mathbf{dv}/2$ selon cette projection.

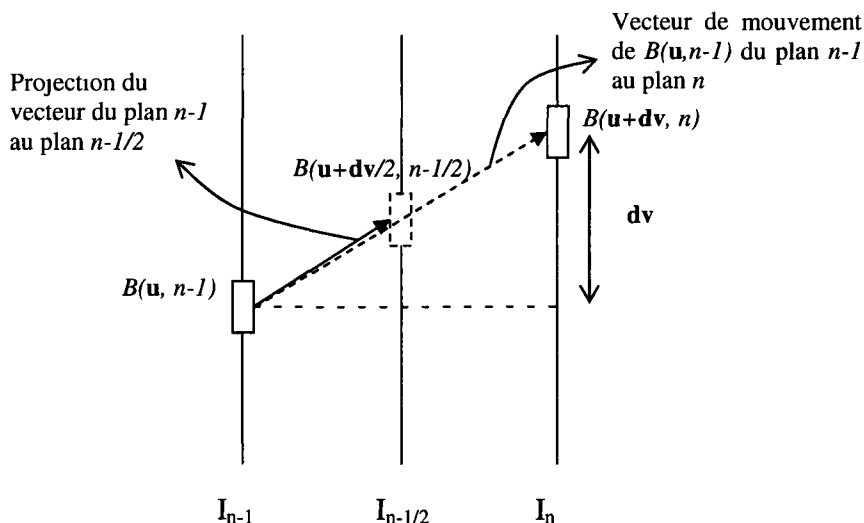


Figure 2.1 Projection le long du mouvement

Généralement

$$\mathbf{IMV}\left(\mathbf{u} + \frac{\mathbf{dv}}{2}, n - \frac{1}{2}\right) = \frac{\mathbf{IMV}(\mathbf{u}, n-1)}{2} = \frac{\mathbf{dv}}{2} \quad (2.1)$$

En réalité, cette projection le long du mouvement génère deux problèmes

- Il y a des régions dans la trame d'interpolation où aucun vecteur de mouvement ne passe. Cela crée des trous dans le champ des vecteurs de mouvement de trame à l'interpolation. Le problème de « trou » (Figure 2 2a)
- Il y a des régions où plusieurs vecteurs de mouvement passent. Ce fait constitue le problème de « chevauchement » (Figure 2 2b). Dans ce cas, on devra choisir un vecteur entre eux pour l'attribuer au champ des vecteurs de mouvement.

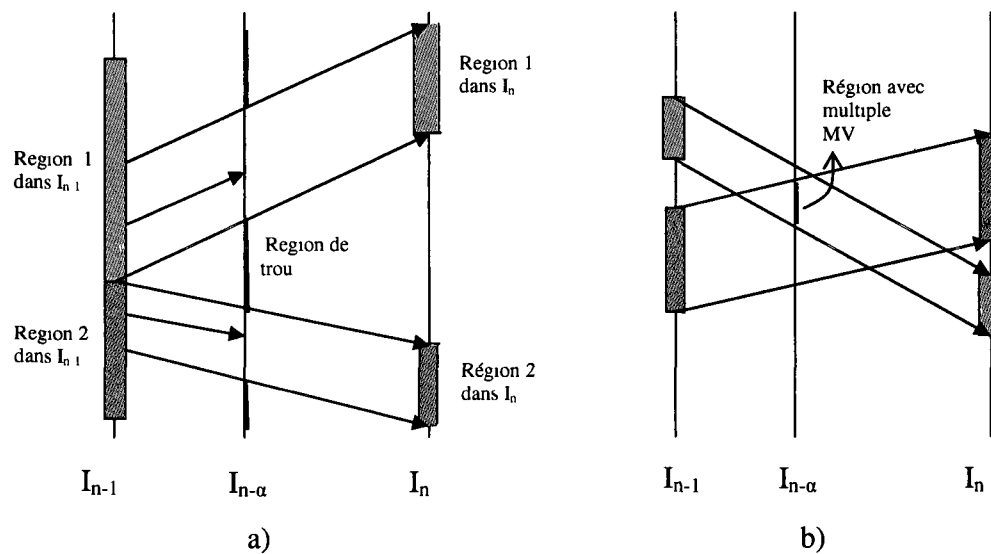


Figure 2 2 Phénomène de « trou » (a) et de « chevauchement » (b)

Le problème de chevauchement est beaucoup plus simple que celui de trou. Jeon *et al.* dans [Jeon *et al.*, 2003] dénouent le problème de chevauchement en choisissant le vecteur de mouvement ayant une moyenne des différences absolues, MAD, minimale entre des vecteurs de mouvement passant le bloc considéré. Leur solution pour le problème de trou était plus compliquée en cherchant quatre points plus appropriés dans ses voisinages, comme c'est illustré à la figure 2 3. Ensuite, le vecteur de mouvement final au trou est remplacé par la moyenne pondérée de ces quatre vecteurs de mouvement.

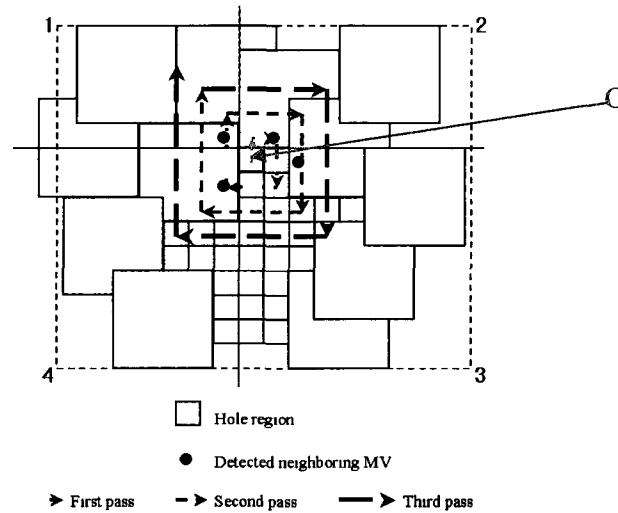


Figure 2 3 Résolution du problème de trou de [Jeon *et al*, 2003] à l'aide d'une recherche en spirale autour du point « O » appartenant à une région de trou pour chercher quatre points plus appropriés dans ses voisinages

Kaup et Aach [Kaup et Aach, 1994] suggèrent une interpolation spatiale. Cependant, cette approche demande des opérations compliquées, car le voisinage d'un trou peut encore contenir d'autres trous. Dans un cas limité pour l'arrière-plan statique, Schutten et De Haan [Schutten et De Haan, 1994] détectent les zones de recouvrement et découverture pour résoudre le problème de trou ainsi que pour réduire l'effet de halo. Cette technique n'est pas efficace si la zone d'occlusion est large ou si l'arrière-plan est mobile.

La projection le long du mouvement ne produit pas seulement un problème de trou et de chevauchement, mais aussi un problème de précision, car $u + \frac{dv}{2}$ n'est pas nécessairement un entier.

[Ling *et al*, 2008] proposent une autre technique pour résoudre le problème de chevauchement dans la projection le long du mouvement. Les auteurs calculent le vecteur de mouvement MV_a d'un bloc à la trame interpolée de la manière suivante :

$$MV_a = \frac{\sum_{t=1}^N mv_t}{N} \quad (2.2)$$

où \mathbf{mv}_1 est le vecteur de mouvement du pixel dans le bloc réel et dans la trajectoire de mouvement N est le nombre de pixels dans le bloc effectif sur la trajectoire de mouvement adoptée Le vecteur de mouvement qui domine un plus grand secteur sera alors le plus important dans le calcul de \mathbf{MV}_a Le résultat \mathbf{MV}_a peut être considéré comme le meilleur candidat de tout le bloc dans la trajectoire de mouvement Pour obtenir une meilleure performance, les auteurs proposent un raffinement sur le vecteur de mouvement obtenu dans une petite région de recherche de 2 pixels en tenant compte des trois SAD entre les images I_n , $I_{n-1/2}$ et I_{n-1} Cependant, cette façon de calculer la moyenne de tous les vecteurs de mouvement dans la trajectoire de mouvement risque d'attribuer un vecteur de mouvement irréal au bloc à la trame interpolée

Étant donné la grande complexité de calcul et la résolution du problème de trou, plusieurs recherches évitent la projection des vecteurs de mouvement le long du mouvement Les techniques présentées dans la section 2.2.2 suivante choisissent une autre façon de projeter les vecteurs de mouvement en utilisant la transposition latérale

2.2.2 Transposition latérale

Soient $\mathbf{dv1}$, $\mathbf{dv2}$ les vecteurs de mouvement respectifs de $B(\mathbf{u}, n-1)$ et de $B(\mathbf{u} - \frac{\mathbf{dv2}}{2}, n-1)$ à la figure 2.4

Une façon plus simple d'obtenir le vecteur de mouvement pour la trame d'interpolation $I_{n-1/2}$, est de transposer latéralement les vecteurs de mouvement de I_{n-1} au $I_{n-1/2}$ C'est-à-dire que le vecteur de mouvement du bloc aux coordonnées \mathbf{u} dans le plan $n-1/2$ sera égal à la moitié du vecteur de mouvement $\mathbf{dv1}$ du bloc à la même coordonnée \mathbf{u} dans le plan $n-1$

$$\mathbf{IMV}(\mathbf{u}, n - \frac{1}{2}) = \frac{\mathbf{IMV}(\mathbf{u}, n-1)}{2} = \frac{\mathbf{dv1}}{2} \quad (2.3)$$

bien que le vrai vecteur de mouvement du $B(\mathbf{u}, n-1/2)$ pourrait être égal à $\mathbf{MV}(\mathbf{u} - \frac{\mathbf{dv2}}{2}, n-1)$

ou $\mathbf{MV}(\mathbf{u} + \frac{\mathbf{dv2}}{2}, n)$

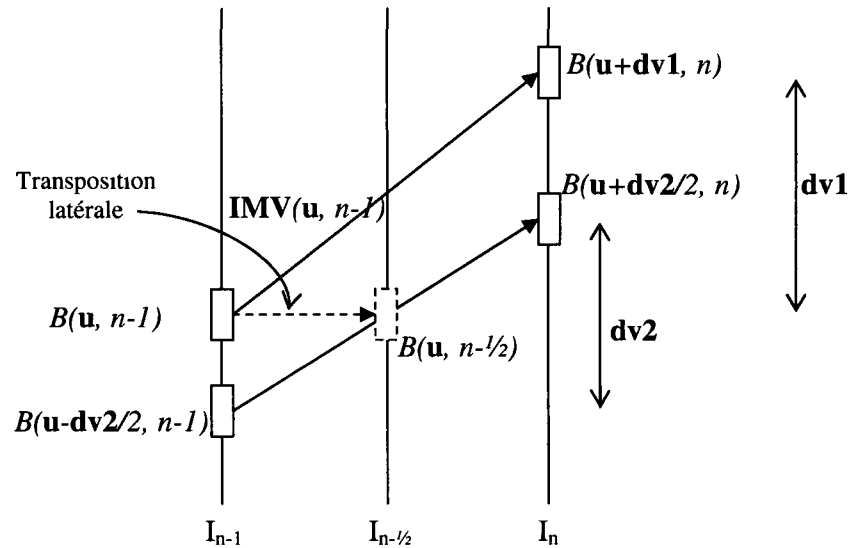


Figure 2 4 Transposition latérale du vecteur de mouvement du bloc $B(\mathbf{u}, n-1)$
Le vecteur de mouvement du bloc $B(\mathbf{u}, n-1/2)$ attribué un mouvement de $d\mathbf{v}1/2$

Si $B(\mathbf{u}, n-1)$ et $B(\mathbf{u} - \frac{d\mathbf{v}2}{2}, n-1)$ sont dans la même région, par exemple, même arrière-plan

ou même objet, alors $IMV(\mathbf{u}, n-1)$ a une grande chance d'être égale à $IMV(\mathbf{u} - \frac{d\mathbf{v}2}{2}, n-1)$

Dans ce cas, la formule (2.3) donne le vrai vecteur de mouvement au bloc $B(\mathbf{u}, n-1/2)$. Cependant, dans les cas où ces blocs appartiennent à deux différentes régions de mouvement ou à cause d'une estimation fautive de mouvement, $IMV(\mathbf{u}, n-1)$ est différente de $IMV(\mathbf{u} - \frac{d\mathbf{v}2}{2}, n-1)$. Ce fait peut résulter en une cassure au bord de l'objet en mouvement ou

en un artefact de bloc. De plus, l'artefact au bord de l'objet en mouvement provoque éventuellement l'effet de halo.

L'association de $IMV(\mathbf{u}, n-1)$ pour $B(\mathbf{u}, n-1/2)$ peut procurer des vecteurs de mouvement non exacts, surtout dans la zone d'occlusion.

2.2.3 Améliorations existantes

Dans le but d'améliorer les deux précédents principes pour la conversion du vecteur de mouvement existant à celui de l'image interpolée, plusieurs publications proposent de chercher un vecteur de mouvement du bloc $B(u, n-1/2)$ le plus ressemblant possible à celui de $IMV(u - \frac{dv2}{2}, n-1)$

Les techniques de cette classe consistent à raffiner les vecteurs de mouvement du plan I_n et/ou I_{n-1} . Ensuite, ces vecteurs seront projetés latéralement du plan $n-1$ et/ou du plan n au plan interpolé $n-1/2$.

De plus, quelques techniques dans cette classe, qui peuvent réduire l'artefact de bloc et l'effet de halo, seront présentées dans la présente sous-section

a Réduction de l'artefact de bloc

Comme raffinement possible, plusieurs auteurs essayent de perfectionner l'étape d'estimation de mouvement entre deux trames d'entrée à l'encodeur afin d'avoir de bons vecteurs de mouvement projetés à la trame interpolée dans le décodeur. Ha *et al* [Ha *et al*, 2004] fournissent une estimation de mouvement avec le recouvrement de différente taille. Ces auteurs modifient également la technique de calcul de la somme des différences absolues (*Sum Absolute Difference*, SAD) afin d'obtenir un meilleur résultat de l'estimation de mouvement. Fujiwara et Taguchi [Fujiwara et Taguchi, 2005] profitent également de la taille de bloc qui est adaptative dans leur estimation de mouvement. L'estimation de mouvement est commencée par blocs de 8x8. Ensuite, en fonction de la propriété du mouvement et de la taille des objets en mouvement, la taille finale de bloc sera 8x8 ou fusionnée en 16x16 ou en 32x32. La taille du bloc dans la région de mouvement global sera la plus grande, tandis que la taille du bloc dans la région locale sera plus petite en répondant à la précision de vecteur de mouvement autour de la bordure de l'objet en mouvement. Enfin, le champ des vecteurs de mouvement de la trame interpolée sera obtenu par une projection latérale de champ des vecteurs de mouvement de la trame à l'entrée. La figure 2.5 suivante présente la technique

Lee *et al* [Lee *et al*, 2007] développent l'application FRC pour le téléphone portable. Les auteurs essaient de trouver les vrais vecteurs de mouvement en utilisant une zone de recherche

adaptive et une taille de blocs adaptative dans l'étape d'estimation du mouvement Sept classes de tailles de blocs sont utilisées selon les différents niveaux de variété de mouvement La taille de bloc est plus petite pour des objets en mouvement Une plus grande taille est choisie pour un bloc dans l'arrière-plan stationnaire

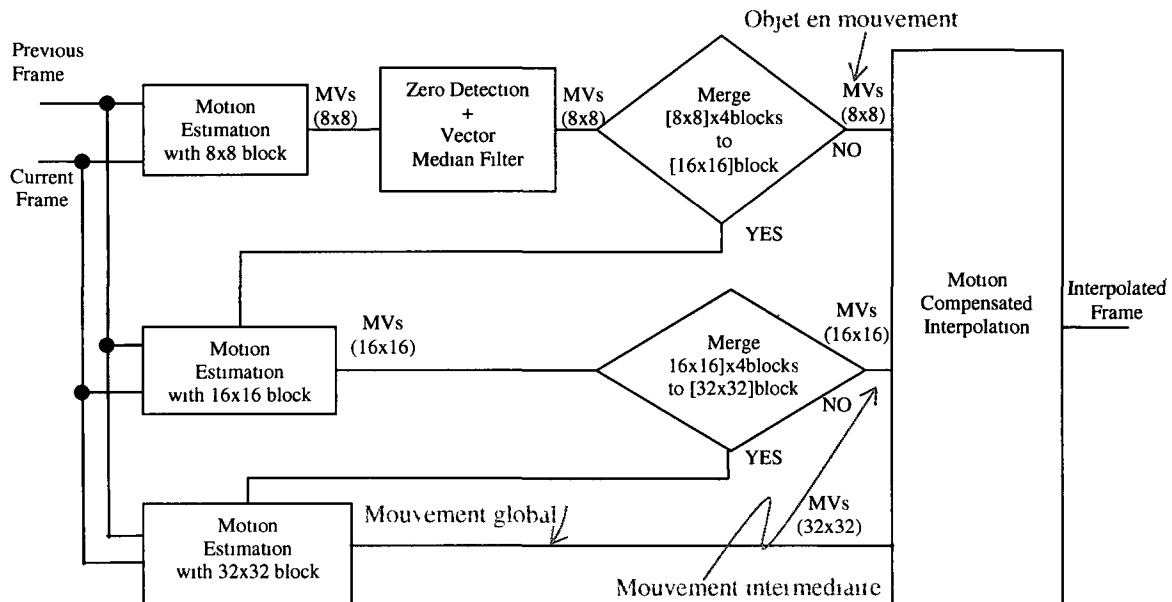


Figure 2 5 Méthode d'estimation de mouvement de [Fujiwara et Taguchi, 2005] utilisant la taille de bloc adaptative pour chaque région de mouvement

Hong *et al* [Hong *et al*, 2006] améliorent l'estimation de mouvement forward dans leur approche Ils analysent la direction dominante du vecteur de mouvement précédent afin d'améliorer l'estimation de mouvement et de faire le post-traitement de vecteur de mouvement Dans la compensation de mouvement, chaque bloc est divisé en quatre différents sous-blocs Chacun des vecteurs de mouvement dans les sous-blocs est comparé avec le mouvement du bloc voisin Après ces comparaisons, le facteur de corrélation est trouvé et celui-ci est appliqué à la compensation de mouvement finale

L'amélioration de l'estimation de mouvement entre deux images d'entrée permet de réduire l'effet de bloc dans l'arrière-plan et autour de l'objet en mouvement Cependant, ces techniques ignorent le problème d'occlusion un problème potentiel de l'interpolation de trame Ces techniques n'arrivent donc pas à réduire l'effet de halo

b Réduction de l'effet de halo

Pelagotti et Schutten [Pelagotti et Schutten, 2006] essaient de réduire le halo dans l'interpolation à compensation de mouvement. Pour ce faire, les auteurs utilisent une moyenne pondérée des interpolations individuelles. Chaque interpolation fournie par le vecteur de mouvement dans un groupe avoisinant se base sur le filtrage médian des interpolations forward, backward et de la valeur moyenne des deux images existantes. Le vecteur de mouvement utilisé dans cette approche est le résultat d'une estimation de mouvement forward basée-bloc de l'image I_n à I_{n-1} . La pondération spécifique d'une interpolation individuelle est la fonction d'une mesure de fiabilité basée sur l'erreur de l'intensité et/ou du pourcentage d'occurrence (de la fréquence d'occurrence relative) du vecteur de mouvement aux alentours. La détection de contour de vecteur de mouvement est également prévue dans le but de réduire la complexité. Il y a quelques inconvénients possibles dans la proposition : le vecteur de mouvement basé bloc peut créer l'effet de bloc, un seul vecteur de mouvement forward pour un bloc peut être insuffisant dans la partie d'occlusion de l'image et enfin, la moyenne des intensités des pixels peut brouiller l'image interpolée dans les régions de l'occlusion ou produire l'effet de dédoublement d'image lorsque le vecteur de mouvement est assez grand.

En se basant sur l'analyse d'histogrammes des vecteurs de mouvement de trame entière, Lee *et al*, [Lee *et al*, 2002] cherchent les vecteurs horizontal et vertical du mouvement global à partir des vecteurs de mouvement locaux. Les auteurs classifient ensuite un bloc de mouvement local ou global en comparant le module du vecteur de mouvement considéré au vecteur de mouvement global. Dans le cas des blocs des vecteurs de mouvement global, l'interpolation se fait selon la méthode conventionnelle comme dans [LeDinh, 2003, De Haan, 2000]. Par contre, dans le cas des blocs des vecteurs de mouvement locaux, un filtre médian est appliqué sur les pixels interpolés avoisinants pour déterminer la valeur médiane d'intensité des pixels respectivement dans chaque trame I_n et I_{n-1} . Et enfin, le pixel final d'interpolation est calculé par la moyenne des deux valeurs médianes. La figure 2.6 démontre le processus de cette technique. Cette technique d'interpolation est donc une combinaison entre les interpolations individuelles dans chaque trame I_n , I_{n-1} et l'interpolation linéaire dans les régions d'objet en mouvement. Selon ces auteurs, la technique sert à éviter le cas où un faux vecteur de mouvement causerait l'artefact de bloc. Cette approche est bonne pour les régions

de mouvement global. Pourtant, elle ne règle pas le problème d'occlusion et le résultat sera flou à cause de la façon d'interpoler dans les régions locales.

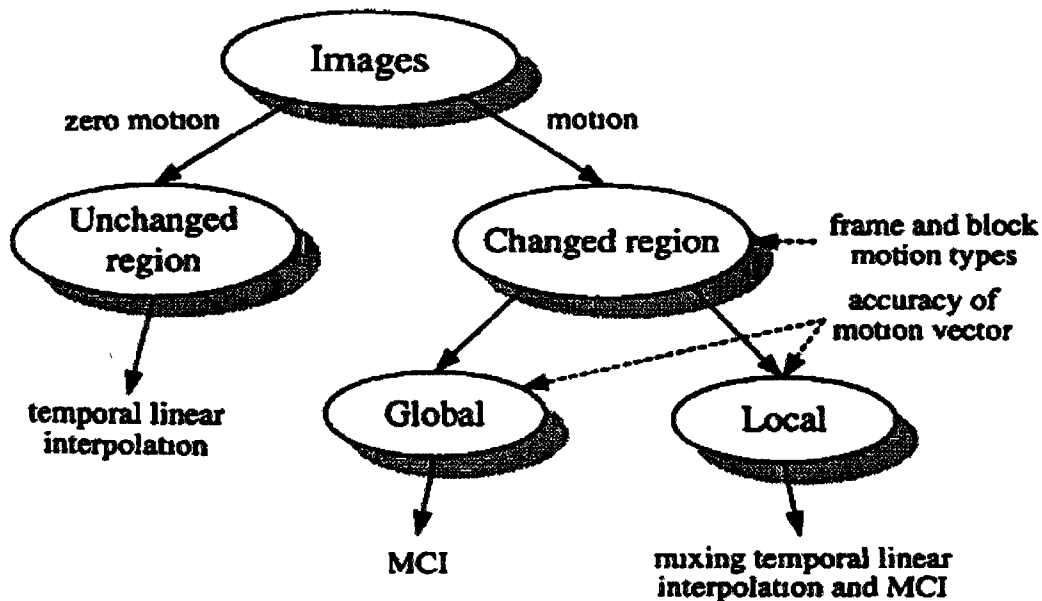


Figure 2.6 Compensation adaptative selon la classification de mouvement [Lee *et al*, 2002]

Par le raffinement hiérarchique des vecteurs de mouvement avec différentes tailles de bloc, dont 4x4 est la dimension minimale, les auteurs de [Huang et Nguyen, 2008] réduisent les problèmes de cassure de bordure et de la déformation des structures dans une trame interpolée. Ce résultat est obtenu par l'analyse de la distribution des énergies résiduelles et par la fusion effective des blocs qui ont des vecteurs de mouvement non fiables. Les informations de chrominance sont aussi utilisées dans cette méthode ce qui crée un coût additionnel à l'algorithme.

[Huang et Nguyen, (2009)] améliorent le travail dans [Huang et Nguyen, (2008)] pour les cas où les vecteurs de mouvement non fiables ne peuvent être détectés par la technique de [Huang et Nguyen, (2008)] en compensation du mouvement d'interpolation de trames. La méthode proposée élimine les vecteurs de mouvement aberrants en calculant la corrélation des vecteurs de mouvement dans leur voisinage local. Ensuite, un filtrage moyen adaptatif s'applique sur les vecteurs de mouvement. Les auteurs proposent également une compensation de mouvement adaptative pour les zones d'occlusion basées sur l'analyse de leur distribution de

mouvement aux alentours Les techniques de [Huang et Nguyen, 2008] et [Huang et Nguyen, 2009] sont assez compliquées en comparaison avec les autres techniques de cette classe

En réduisant la taille du bloc, les auteurs dans les publications citées ne fournissent que les techniques de raffinement partiel des vecteurs de mouvement, et ce, sur les trames originales, mais pas directement aux vecteurs de mouvement de la trame d'interpolation considérée Toutefois, le champ des vecteurs de mouvement demeure basé bloc et le vecteur de mouvement ne peut pas être précis à la bordure de l'objet en mouvement Ces techniques proposées ne réduisent donc pas complètement l'effet de bloc ni celui de halo Par contre, la complexité de la plupart des approches dans cette classe est assez simple

La figure 2.7 suivante illustre l'effet de bloc mentionné dans l'image et l'interpolation finale de [Lee *et al.*, 2007] Dans cette référence, les auteurs proposent même une estimation de mouvement utilisant les tailles de bloc adaptatives et la fenêtre de recherche en fonction du niveau de l'activité du mouvement

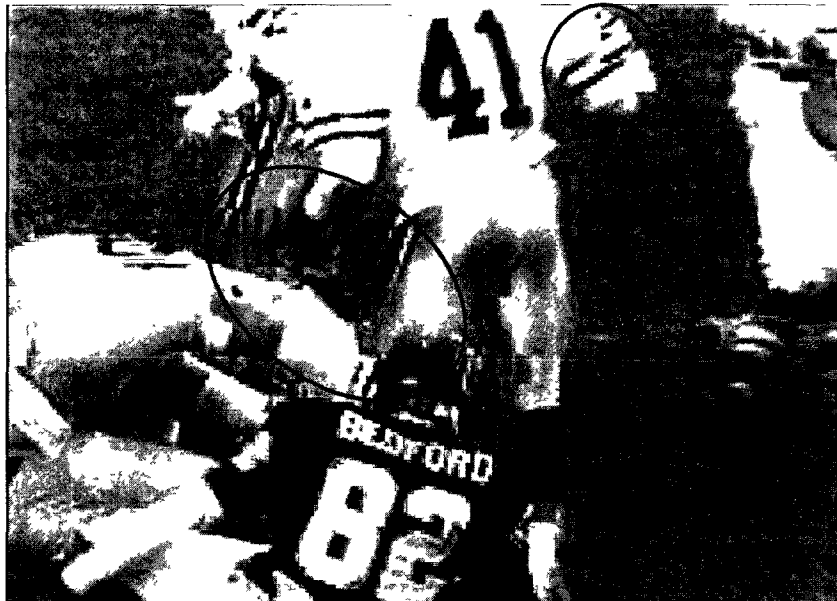


Figure 2.7 Artefact de bloc visible dans les zones indiquées en rouge d'un des résultats de [Lee *et al.*, 2007]

Finalement, comme dans la plupart des cas, les auteurs de [Ha *et al.*, 2004, Lee *et al.*, 2007, Fujiwara et Taguchi, 2005, Hong *et al.*, 2006] ne mentionnent que le doublage de fréquence

d'images par interpolation dans leurs approches, mais ils ne considèrent pas une conversion de taux d'images générale à interpolations multiples

2.3 Interpolation temporelle utilisant l'estimation bidirectionnelle

Contrairement à la section précédente où le vecteur de mouvement provient d'une estimation unidirectionnelle, i.e. forward ou backward, les approches de cette classe utilisent souvent l'algorithme par appariement de bloc bidirectionnel (*Bidirectional Block-based Motion Estimation*, BDME) pour obtenir le champ de vecteur de mouvement de trame à l'interpolation

BDME divise image à interpoler en des blocs. Un vecteur de mouvement sera calculé et attribué à chacun de ces blocs. Supposons que nous voudrions trouver le vecteur de mouvement pour un bloc, ayant la taille W , aux coordonnées u dans la trame à l'interpolation $I_{n-1/2}$ par un BDME, comme illustré dans la figure 2.8. SZ_{n-1} est une zone de recherche à la trame I_{n-1} et sa zone alignée à la trame I_n est SZ_n . Un appariement de chaque paire de blocs symétriques par rapport à u avec les zones de recherche SZ_{n-1} et SZ_n est exécuté selon la formule suivante

$$IMV(u, n - \frac{1}{2}) = \arg_{dv} \min \sum_{dv \in SZ} |I_n(uW + dv) - I_{n-1}(uW - dv)| \quad (2.4)$$

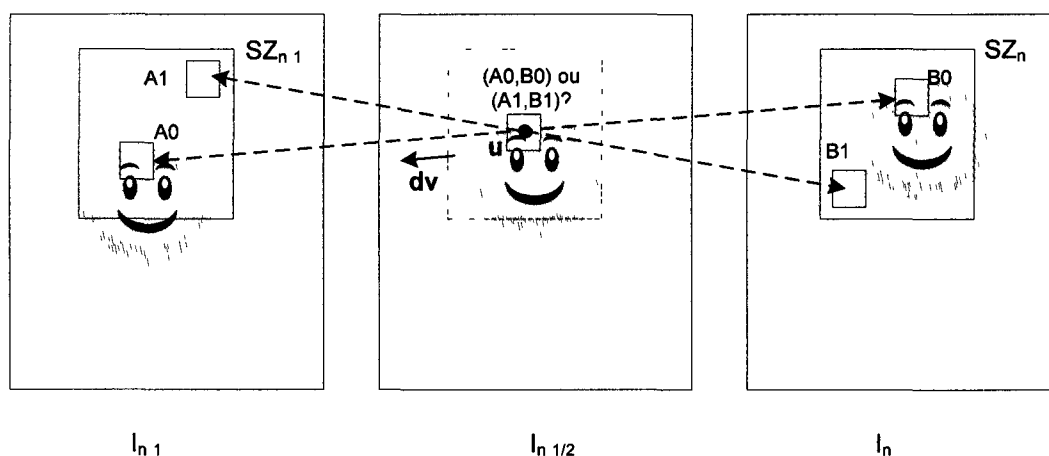


Figure 2.8 Méthode BDME et son ambiguïté éventuelle des vecteurs de mouvement entre (A_0, B_0) ou (A_1, B_1)

Un point fort de BDME est qu'il donne le champ des vecteurs de mouvement au plan d'interpolation. Cela évite les problèmes causés par la projection de vecteur de mouvement en discussion dans les sections 2.2.1 et 2.2.2. Par contre, un point faible de BDME est sa sensibilité aux régions plates ou aux régions de texture. À la présence de quelques facteurs, comme le bruit, le changement d'éclairage, un appariement de deux blocs dans la zone plate aura plus de chance de donner le minimum d'erreurs d'appariement qu'un appariement des objets en mouvement.

La figure 2.8 illustre un cas où BDME risque de donner le meilleur appariement à des paires dans la zone plate. Le vrai vecteur de mouvement doit être le déplacement de l'objet, de A_0 à B_0 . Cependant, on voit aussi plusieurs appariements de pseudo-paires plus identiques que la paire (A_0, B_0) , comme la paire (A_1, B_1) et leurs paires voisines. La figure 2.9 suivante démontre l'effet de ce problème pour la séquence *Flower Garden* en utilisant la BDME conventionnelle.

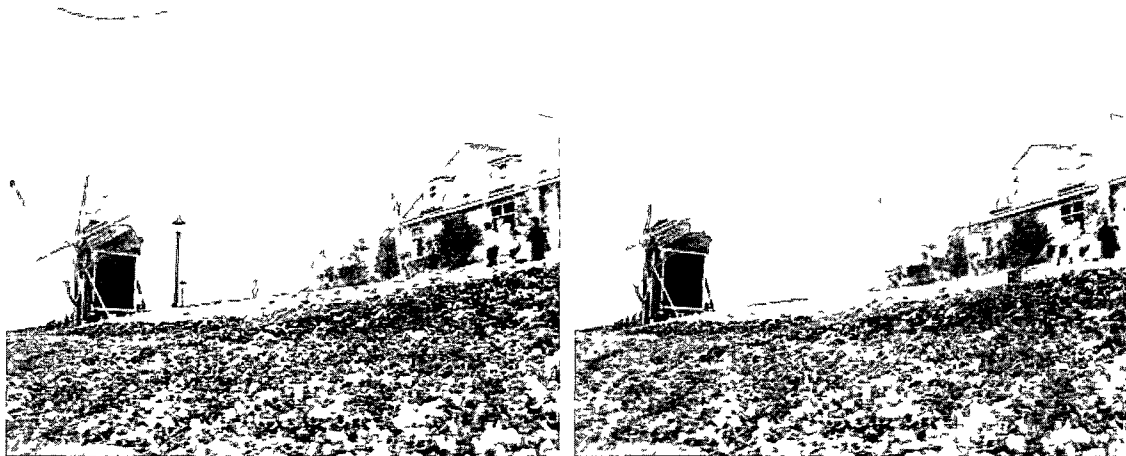


Figure 2.9 Séquence *Flower Garden*. Trame 65^e originale et trame interpolée avec BDME, bloc de dimension 16×16 , zone de recherche $\pm 20 \times \pm 24$

Soit α , une valeur fractionnaire, qui est la distance normalisée entre l'image existante I_n et l'image interpolée souhaitée $I_{n-\alpha}$. Le processus pour déterminer le vecteur de mouvement au plan $I_{n-\alpha}$ dans [Choi *et al*, 2000] est le suivant. Supposons \mathbf{dv} , le vecteur de mouvement d'un bloc \mathbf{b} provenu de BMA forward, et projeté latéralement au plan $I_{n-\alpha}$. Les auteurs raffinent, par la suite, le vecteur de mouvement \mathbf{dv} par un BDME où $\mathbf{dv} \pm \mathbf{d}$ (\mathbf{d} est petit) sont candidats

d'appariement Enfin, un autre BDME est calculé en employant huit vecteurs voisins de bloc considéré au plan $I_{n-\alpha}$ comme ses candidats Les vecteurs de mouvement finaux obtenus par cette méthode sont encore basés-bloc Une compensation de mouvement adaptative moyenne est proposée dans cet article pour réduire l'artefact de bloc

Si les auteurs de [Choi *et al*, 2000] font un BDME après forward BMA, Shi et Raveendran dans [Shi et Raveendran, 2005, USP2005/0265451A1] profitent du vecteur de mouvement de BDME comme le vecteur de mouvement initial pour réaliser une estimation de mouvement forward ou une estimation de mouvement backward selon la région de recouvrement ou de découvrément Pourtant, leur estimation de mouvement est encore basé bloc

Un algorithme d'appariement hiérarchique de blocs est utilisé pour estimer le champ de vecteurs de déplacement dans [Thoma et Bierling, 1989] En analysant les vecteurs de mouvement et l'information d'un détecteur de changement, une segmentation des images en quatre régions objets en mouvement, arrière-plan statique, découvrément d'arrière-plan et recouvrement d'arrière-plan, est exécutée Un filtre spatio-temporel contrôlé par les informations de déplacement et de segmentation, est utilisé pour obtenir par interpolation le nombre d'images requis Néanmoins, la méthode de [Thoma et Bierling, 1989] se base sur l'hypothèse de l'arrière-plan statique

Choi *et al* [Choi *et al*, 2007] améliorent le travail de Choi *et al*, [Choi *et al*, 2000] L'image est divisée en blocs de différentes tailles Les régions homogènes avec des mouvements de translation sont décomposées en blocs de grande dimension, tandis que celles avec des mouvements complexes sont divisées en blocs dont la taille est plus petite Pour réduire l'artefact de bloc autour d'un objet en mouvement, les blocs sont divisés encore en sous-blocs pour reconstruire avec plus de précision les bordures de l'objet Cette technique emploie une opération itérative dans la période de segmentation, elle sera difficilement implémentée en temps réel

Les auteurs de [Kang *et al*, 2007] améliorent l'estimation de mouvement bidirectionnel (BDME) classique Comme la figure 2 10 l'illustre, après avoir calculé BDME classique pour deux blocs originaux, une autre BDME est réalisée pour le bloc supplémentaire chevauchant deux blocs d'origine sur la trame à l'interpolation Ensuite, les sommes des différences

absolues de trois blocs sont comparées afin de choisir le meilleur vecteur de mouvement ayant le minimum d'erreurs d'appariement. Cette méthode améliore la résolution par un facteur de quatre dans la zone de recherche, mais elle ne règle pas le problème de plusieurs blocs ressemblants, un désavantage de la méthode BDME, comme illustré dans la figure 2.8

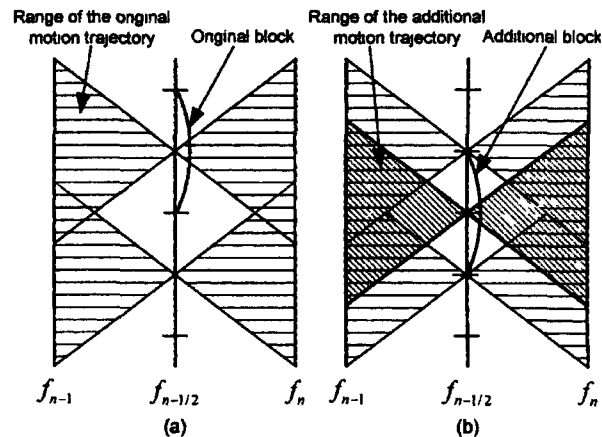


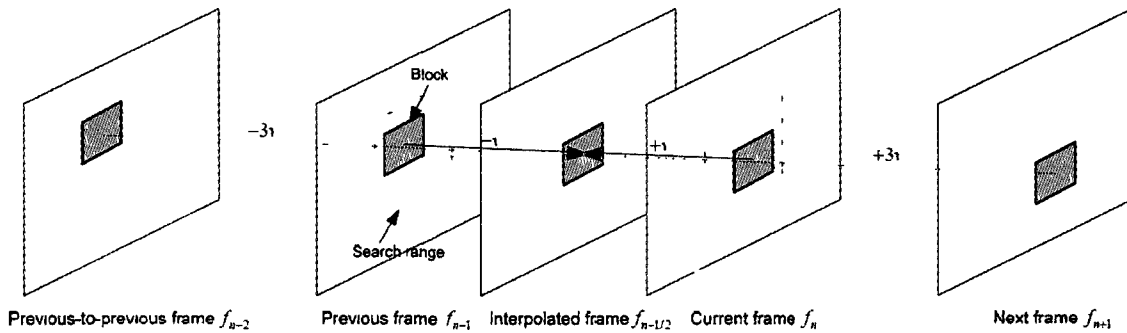
Figure 2.10 (a) BDME classique, (b) BDME avec bloc additionnel

Doublage de résolution de l'estimation par dimension dans (b) (par rapport à (a))

De Haan et Pelagotti dans [De Haan et Pelagotti, 2006, USP7058227B2] et [De Haan et Pelagotti, 2002, USP6487313] supposent que les discontinuités dans le champ des vecteurs de mouvement correspondent aux frontières des objets en mouvement. Les auteurs proposent donc d'utiliser i) la longueur de vecteur de mouvement afin de déterminer les zones d'occlusion, ii) une erreur de calcul supplémentaire de l'interpolation bidirectionnelle utilisant le bloc de petite taille pour augmenter la précision des zones d'occlusion détectées, iii) et le filtrage médian adaptatif pour l'interpolation finale. Toutefois, la longueur de vecteur de mouvement n'est pas nécessairement un paramètre fiable, lorsque les objets sont composés de régions plates et bruitées. Les discontinuités dans le champ des vecteurs de mouvement et la bordure des objets mobiles exigent une certaine précision coûteuse, et le filtrage médian demeure encore une technique ad-hoc de correction, car le vecteur de mouvement n'est pas corrigé efficacement.

Les vecteurs de mouvement finaux des techniques mentionnées auparavant sont encore basés bloc, les vecteurs de mouvement ne sont donc pas précisément adaptés aux bordures de l'objet. Autrement dit, il y a un groupe de pixels d'arrière-plan ayant le même vecteur de mouvement que l'objet en mouvement. Cela produit l'effet de halo autour de l'objet.

L'utilisation de multiples trames d'entrées dans [Bellers *et al.*, 2007, Kang *et al.*, 2008, Sugiyama *et al.*, 2005, Thomas et Burl, 1999, (USP6005639), Burl *et al.*, 2000, (USP60001596), De Haan *et al.*, 2006, (USP7010039), Li, 2007, (USP2007/0121725A1)] donnent plus d'informations pour réduire les artefacts. La figure 2.11 illustre un exemple de cette classe. Cependant, cette approche exige plus de mémoire d'image à stocker et augmente la bande passante entre la mémoire et le processeur.



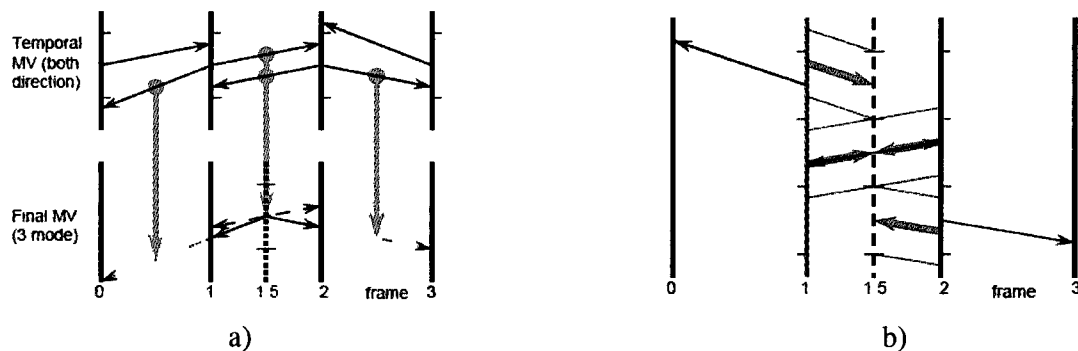
v est le déplacement du bloc considéré de $f_{n-1/2}$ à f_n , $3v$ est le déplacement du bloc considéré de $f_{n-1/2}$ à f_{n+1} , d'une façon similaire, $-v$, $-3v$ de $f_{n-1/2}$ à f_{n-1} et à f_{n-2} respectivement.

Figure 2.11 Amélioration de la méthode BDME classique avec quatre images à l'entrée f_{n-2} , f_{n-1} , f_n et f_{n+1} ([Kang *et al.*, 2008])

Les auteurs de [Bellers *et al.*, 2007] et [De Haan *et al.*, 2006, USP7010039] utilisent trois trames consécutives pour réduire l'artefact halo. Kang *et al.* [Kang *et al.*, 2008] utilisent une méthode BDME adaptative avec quatre trames d'entrée, comme illustre la figure 2.11.

La technique de [Sugiyama *et al.*, 2005] a besoin de quatre trames d'entrée pour réduire le problème d'occlusion. Pour calculer l'estimation de mouvement, les auteurs utilisent la somme des différences absolues normalisée (SAD_{nor}) par l'écart-type du bloc, défini comme la racine carrée de la variance d'un bloc. Leur estimation de mouvement se compose de deux étapes. Pour la première étape, illustrée à la partie supérieure de la figure 2.12 a), ils exécutent

ME forward et ME backward entre les deux trames 1 et 2 afin d'obtenir la somme des différences absolues et le vecteur de mouvement temporel. Pour la deuxième étape, illustrée à la partie inférieure de la figure 2.12 a), dépendamment du minimum de ce SAD_{nor} et des vecteurs de mouvement temporels, un BDME sera effectué dans la région de non-occlusion. Si la différence absolue des vecteurs de mouvement forward (MVf) et backward (MVb), $|MVb - MVf|$, se rapproche de 1, la zone de recherche pour BDME est $z \pm 1$, où z est le vecteur de mouvement moyen de MVb et MVf. Par contre, si $|MVb - MVf|$ est supérieur à 1, la zone de recherche est plus large avec le vecteur de mouvement initial z est égal à 0. Ensuite, une estimation forward qui représente le déplacement de la trame 0 à la trame 1 et une estimation backward qui représente le déplacement de la trame 3 à la trame 2 sont exécutés. En fonction des trois derniers SADs, un des modes d'interpolation forward, backward ou bi-directionnelle sera choisi pour l'interpolation finale.



Partie supérieure de 2.12 a) ME forward et ME backward calculés entre deux trames originales, Partie inférieure de 2.12 a) BDME utilisé lorsque SAD_{nor} est petite et ME forward or ME backward utilisé lorsque SAD_{nor} est grand, b) Interpolation adaptative avec 4 trames en fonction de SAD_{nor}

Figure 2.12 Approche quatre trames de [Sugiyama *et al.*, 2005]

Dans le même but de segmenter l'image en quatre régions, comme [Thoma et Bierling, 1989] Kuo et Kou [Kuo et Kou, 1998] utilisent plus d'images pour cette fin. Si un pixel a un problème de chevauchement, il est remplacé par la moyenne des intensités de ses voisins. Les auteurs résolvent le problème de trou de vecteur de mouvement en raffinant ses vecteurs de mouvement voisins ou en utilisant une interpolation spatiale.

Haan et Pelagotti [De Haan et Pelagotti, 2001, (USP6219436B1)] proposent une deuxième estimation de mouvement basée bloc avec l'erreur à la puissance 4 pour mieux détecter les régions d'occlusion. Le MV, à un pixel donné dans les zones détectées, est par la suite remplacé par celui correspondant à une position décalée. Le calcul des erreurs d'ordre élevé est d'abord coûteux et la substitution des vecteurs de mouvement est une approche ad-hoc. En effet, les auteurs préfèrent encore les mêmes stratégies de filtrage médian pour cacher éventuellement les défauts.

Finalement, même avec un meilleur potentiel de réduction de halo, les solutions multi-frames sont coûteuses. Elles ne sont pas prises en compte dans notre approche.

Les techniques de conversion du taux d'image étudiées sont résumées dans le tableau suivant. Leur complexité de calcul est aussi indiquée.

Tableau 2 1 Résumé des techniques de FRC étudiées

| Groupe | ME | Caractéristiques | Références | Complexité |
|--------|------------------------------------------------|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| # 1 | BMA (basé bloc) | Deux trames Transposition latérale | [Fujiwara et Taguchi, 2005, Huang et Nguyen, 2008, Lee <i>et al</i> , 2007, Hong, Park et Berkeley, 2006, Huang et Nguyen, 2009, Ha <i>et al</i> , 2004, Pelagotti et Schutten, 2006,(USP7039109B2), Lee <i>et al</i> , 2002] | Faible-Moyenne |
| | | Deux trames Projection du MV le long du mouvement | [Kaup et Aach, 1994, Schutten et De Hann, 1994, Ling <i>et al</i> , 2008, Jeon <i>et al</i> , 2003] | Moyenne-Grande |
| | | Deux trames Segmentation d'intensité | [Thomas et Bierling, 1989, De Haan et Pelagotti, 2006, (USP7058227B2), De Haan et Pelagotti, 2002, (USP6487313)] | Moyenne-Grande |
| # 2 | BDME ou (BDME et BMA) (basés bloc) | Deux trames | [Choi <i>et al</i> , 2000, Shi et Raveendran, 2005, (USP2005/0265451A1), Kang <i>et al</i> , 2007, Choi <i>et al</i> , 2007] | Moyenne-Grande |
| | | Multi-trames | [Bellers <i>et al</i> , 2007, Kang <i>et al</i> , 2008, Sugiyama <i>et al</i> , 2005, Thomas et Burl, 1999, (USP6005639), De Haan <i>et al</i> , 2006(USP7010039), Kuo et Kou, 1998, Li, 2007, (2007/0121725A1)] | Très grande |

| Groupe | ME | Caractéristiques | Références | Complexité |
|--------|-------------------------------------|------------------|--------------------------------------------------------------------------|-----------------------|
| #3 | ME par maillages combine BMA | Basé bloc | [Choi <i>et al</i> , 2006] | Réalisation difficile |
| | ME basé sur l'objet | Basé objet | [Blanchfield <i>et al</i> , 2006, Wang et Vincent, 2003, (USP6625333B1)] | |
| | ME basé sur la corrélation de phase | Basé bloc | [Burl <i>et al</i> , 2000, (USP6001596)] | |
| | ME flux optique, BMA | Itérative | [Krishnamurthy, Woods et Moulin (1999)], [Choi, et <i>al</i> ,(2007)] | |

2.4 Conclusion du chapitre et recherche proposée

En partant avec l'algorithme populaire d'appariement des blocs, l'étude de l'état de l'art nous permet de dégager les conclusions suivantes

Dans le but d'obtenir les vecteurs de mouvement dans le plan interpolé, les deux méthodes étudiées, la projection le long du mouvement et la transposition latérale, comportent chacune ses propres inconvénients, ainsi que l'estimation de mouvement bidirectionnelle qui peut procurer pourtant un unique vecteur de mouvement à chaque pixel interpolé. Quoiqu'il en soit, les améliorations existantes résultent encore en deux défauts notables et majeurs dans l'image interpolée : l'effet de bloc et l'effet de halo.

Comme l'algorithme d'estimation de mouvement initial choisi basé-bloc est plus ou moins inconsistant avec la nature changeante de l'image, l'effet de bloc se produit souvent aux contours d'objets ou aux bordures des changements de vecteur de mouvement dans le champ des vecteurs de mouvement.

L'effet de halo vient aussi de l'algorithme d'estimation de mouvement dont le modèle du vecteur de mouvement n'est plus valide dans les régions d'occlusion. Le vecteur de mouvement forward ne peut être valide que dans certaines parties de découvrément. Et d'une façon similaire, le vecteur de mouvement backward pour le recouvrement. Quant au BDME, le vecteur de mouvement peut être totalement faux dans ces régions.

Comme le modèle du vecteur de mouvement dans ces régions devient fautif, l'utilisation des erreurs SAD de prédiction de mouvement pour détecter le phénomène halo devient tout simplement non valable. C'est aussi la raison pour laquelle plusieurs auteurs font appel aux techniques de filtrage d'image pour masquer les éventuels défauts.

Dans ce travail, la recherche proposée comprend alors quatre volets :

- D'abord, pour annuler l'effet de bloc, il est intéressant d'avoir une estimation de mouvement à la résolution de pixel tout en conservant l'algorithme populaire d'appariement des blocs BME, *Block Matching Estimation*, comme estimation préliminaire.
- Ensuite, pour réduire l'effet de halo, une technique de raffinement des vecteurs de mouvement sera proposée, et ce, sans recours à une détection explicite des occlusions. Une nouvelle version d'interpolation d'image sera aussi suggérée.
- Pour la HDTV qui requiert une grande complexité de calcul, l'extension des deux précédents volets sera aussi proposée dans le contexte d'une décomposition hiérarchique.
- Finalement, plusieurs post-traitements permettant les améliorations seront aussi complétés.

Le prochain chapitre concernant principalement la SDTV sera réservé pour les deux premiers items tandis que le traitement hiérarchique de la HDTV sera au chapitre 4. Quant aux post-traitements, ils seront décrits à chaque chapitre respectif.

Pour fin de compréhension, les figures 2.13 et 2.14 illustrent respectivement les schémas blocs des convertisseurs du taux d'images (FRC) en SDTV et en HDTV dans leur entier. Les parties en traits pointillés ME (BME forward et backward), MVF (Filtrage des vecteurs de mouvement) et *Fallback* (Technique de repli) ne seront pas discutées dans le présent travail.

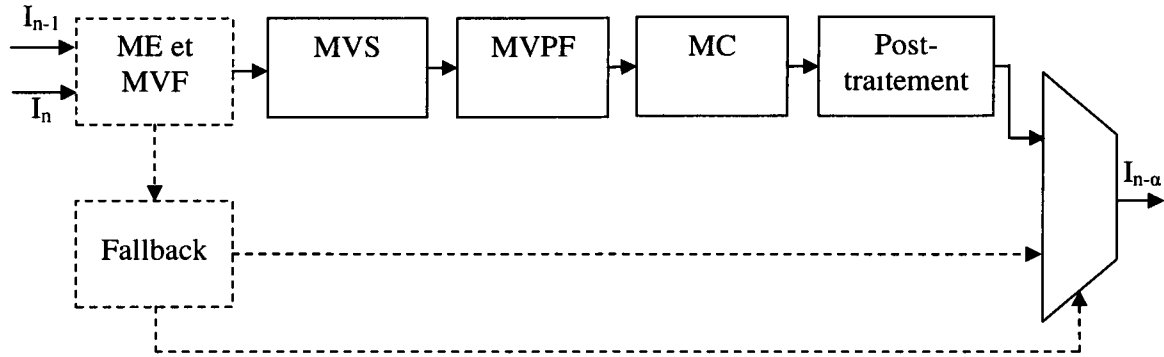


Figure 2 13 Convertisseur FRC complet pour SDTV

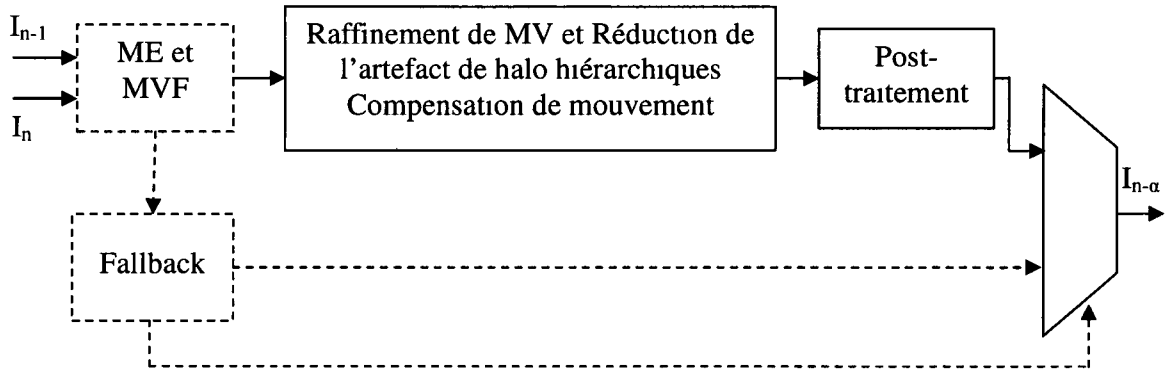


Figure 2 14 Convertisseur FRC complet pour HDTV (à multi -niveaux hiérarchiques)

3 INTERPOLATION TEMPORELLE PROPOSÉE POUR SDTV - ESTIMATION DE MOUVEMENT BASÉE PIXEL ET HALO RÉDUIT

Le présent chapitre sera organisé comme suit

La section 3.1 constituera l'avant-propos pour l'article publié selon le standard défini par l'université. La section 3.2, intitulée Interpolation d'image avec l'estimation de mouvement basée pixel et réduction de halo sera réservée à l'article accepté. La section 3.3 présentera quelques traitements supplémentaires sur l'image obtenue à la section 3.2. Finalement, la section 3.4 démontrera les résultats de ce chapitre.

3.1 Avant propos

Auteurs et affiliation

Thi Thuy Ha Tran étudiante au doctorat, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique

Chon-Tam LeDinh professeur, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique

Date d'acceptation 19 mai 2010 par le professeur Gerard de Haan en tant que l'éditeur associé

État de l'acceptation version finale publiée

Date de prépublication sur le site web de IEEE 12 Juillet 2010

Revue *IEEE Journal of Selected Topics in Signal Processing JSTSP*

Référence [Tran, T H et LeDinh C T (avril 2011), Frame Rate Converter with Pixel-based Motion Vectors Selection and Halo Reduction using Preliminary Interpolation, *IEEE Journal of Selected Topics in Signal Processing*, IEEE JSTSP, *Special Issue on Recent Advances in Video Processing for Consumer Displays RAVPCD*]

Titre français Interpolation d'image avec l'estimation de mouvement basée pixel et réduction de halo

Contribution du document

Cet article contribue à la thèse en élaborant les nouvelles approches de base mais fondamentales pour l'interpolation d'image, à savoir, une nouvelle estimation de mouvement à la résolution en pixel basée sur l'algorithme d'appariement de blocs (BMA) classique, une réduction de halo sans détection des occlusions et une nouvelle version pour l'interpolation multiple d'images

Résumé

Dans le présent article, après une étude de l'état de l'art, une nouvelle méthode pour la conversion du taux d'image est proposée pour réduire les artefacts de bloc, de halo et de scintillement

Pour l'artefact de bloc, une sélection du vecteur de mouvement à la résolution du pixel est suggérée en se basant sur les vecteurs de mouvement environnants estimés à partir de l'algorithme BMA classique. Le critère de sélection est basé sur la minimisation d'une erreur pondérée composée des deux approches de l'estimation de mouvement : celle de l'estimation bidirectionnelle BDME et celle de BMA.

Pour l'effet du halo, l'algorithme suggéré comprend deux étapes. Il s'agit d'abord d'une interpolation préliminaire de l'image à partir des vecteurs de mouvement sélectionnés. Ensuite, un raffinement de vecteur de mouvement est effectué en se basant sur une fenêtre glissante mais adaptée à la forme. L'adaptation de la forme est basée à son tour sur deux paramètres locaux : les intensités moyennes de l'image interpolée et les vecteurs de mouvement dans la fenêtre glissante par rapport à ceux du pixel considéré. L'algorithme suggéré appliqué à tout pixel dans l'image ne recourt à aucune détection explicite des zones de recouvrement ou de découvrément.

Pour l'effet de scintillement, une nouvelle version d'interpolation d'images est aussi proposée. Il s'agit d'abord du facteur de fiabilité, qui est le rapport du nombre des pixels locaux favorisant une interpolation avec le vecteur de mouvement backward sur la dimension totale d'une fenêtre glissante. Vient ensuite une nouvelle forme plus robuste pour une

interpolation multiple des images. L'interpolation suggérée combine à la fois le facteur de fiabilité et la position de l'image interpolée.

Les résultats expérimentaux démontrent une performance supérieure objectivement et subjectivement par rapport à quelques méthodes existantes.

Norme universitaire : Les figures, formules et références dans l'article sont renumérotées selon le standard de Université de Sherbrooke.

3.2 Interpolation d'image avec l'estimation de mouvement basée pixel et réduction de halo

Frame Rate Converter with Pixel-based Motion Vectors Selection and Halo Reduction using Preliminary Interpolation

Thuy-Ha thi Tran and Chon-Tam LeDinh, *Member, IEEE*

Abstract — A new two-image based method for frame rate conversion is proposed to reduce blocking, halo and flickering artifacts. For blocking effects, a pixel-based motion vector (MV) selection is suggested based on neighboring block-based motion vectors. For halo reduction, after a preliminary image interpolation, MV at each pixel is re-estimated using the interpolated intensities and the MV of the current pixel and its neighbors constrained in adaptive sliding windows. Experimental results showed that the proposed method outperforms objectively and subjectively in comparison with some existing interpolation techniques.

Index Terms — Pixel-based Motion Vector Selection, Halo Reduction, Intensity Adaptive Windowing, Motion Adaptive Windowing, Frame Rate Converter, Real time Processing.

3.2.1 Introduction

In recent years frame rate up-conversion (FRUC) or simply frame rate conversion (FRC) has become an important technique for various film/video conversions, temporal video formats

coding and technology displays. Many motion estimation (ME) and motion-compensated interpolation (MCI) algorithms, forming the basis of typical FRC solutions, have been proposed for motion judder reduction [De Haan *et al*, 1993, Krishnamurthy *et al*, 1999]. Hierarchical Block Matching Motion Estimation was presented in [De Haan *et al*, 1993, Jeon *et al*, 2003] for complexity reduction over classical Block Matching Motion Estimation (BME). The FRC methods can be further classified into two approaches. The first approach estimates the motion vector (MV) between the previous and the current frame and then generates the desired interpolated frames [De Haan *et al*, 1993, Thoma et Bierling, 1989, Hilman, *et al*, 2000, Jeon *et al*, 2003, Choi *et al*, 2007]. The second approach working in the decoding enhancement context utilizes the received MV provided by various image compression standards such as MPEG1, MPEG2 [Krishnamurthy *et al*, 1999, Huang et Nguyen, 2009].

Among various ME techniques, the block-matching algorithm (BMA) frequently used in temporal prediction coding is the most popular for low computation and ease of implementation [Castagno *et al*, 1996]. Considering that a single motion vector is invariably assigned to a given block in a reference image, BMA can give rise to various artifacts such as blocking effect, but also to holes and overlapped areas in the interpolated image, when remapping the obtained MV. Various algorithms have been proposed to reduce these artifacts [Thoma, 1989], [Hilman *et al*, 2000]. For avoiding holes and overlaps phenomena, block-based bi-directional ME (BDME) proposed in [Choi *et al*, 2000], [Kang *et al*, 2007] is a capable solution. BDME estimates a bidirectional motion vector of the reference block in the interpolated image using geometrical similitude relationship between the previous and the current input image. It provides a MV for each pixel or block in the interpolated image. In order to reduce blocking artifacts, overlapped block motion compensation (OBMC) was suggested [Hilman *et al*, 2000], [Choi *et al*, 2007]. OBMC can be incorporated in Motion Compensated Interpolation (MCI) by appropriately combining interpolated results from neighboring block MV. A similar approach was generalized with weighted-adaptive MCI [Lee *et al*, 2003] in which the weighting or accuracy function was determined by using predicted ME errors. In [Lee *et al*, 2002], the suggested MCI was based on median filtering of various pixel intensity candidates which were obtained from neighboring block motion vector.

trajectories passing through the interpolated pixel location. The re-sampling of the MV field in smaller blocks to smooth motion vectors was also proposed [Dane et Nguyen, 2004].

The benefit of such an approach is to increase MV field resolution. The halo effect caused by erroneous estimated MV in occlusion regions is also an important artifact in FRC [Thoma, 1989]. Existing halo reduction algorithms can be classified into two categories using two or more successive input frames. Multiple frame based analysis [De Haan, 2000]-[Kang *et al* , 2008] can provide greater potential for efficient halo reduction. However, in the present case, a two-frame based solution [Thoma, 1989], [Jeon *et al* , 2003]-[Lee *et al* , 2002] is preferred for cost consideration. Most FRC algorithms with an emphasis on halo-artifact reduction are based respectively on two aspects: covered or uncovered (C/U) area detection and adaptive forward and backward MCI while assuming that estimated forward or backward MVs are adequate for image interpolation. Covered and uncovered area detection is provided generally by forward and backward prediction errors or MV boundaries. However, in occlusion regions, motion compensated prediction models are not necessarily valid, since none of the classical forward, backward and bidirectional ME between two existing images can correctly estimate the true MV. Therefore, C/U area detection provided by associated prediction errors can be difficult to approximate with practical accuracy. In a teleconference application, where the background is stationary, the C/U detection as shown in [Thoma, 1989] can be based solely on foreground MV. However, in more generic applications, weighted-adaptive MCI or median filtering MCI has been proposed [De Haan, 2000] to blur the halo artifact in the detected C/U area. Recently, in classified occlusion regions, median MV filtering [Astola *et al* , 1990] in a spiral scanning from the left-top and clockwise direction was suggested for re-estimated MV [Huang et Nguyen, 2009].

Frame-based iterative techniques and object-based segmentations are software-oriented concepts for artifact reduction [Choi *et al* , 2007], [Blume *et al* , 2002], [Blanchfield *et al* , 2006] but generally, their hardware implementation is rather difficult and becomes another challenge altogether for a real-time FRC.

In this paper, a new image interpolation scheme is proposed to overcome the difficulties outlined previously. The suggested scheme is based mainly on MV estimation and processing between two adjacent frames. Thanks to recent advances in ASIC technology, after

conventional BMA, a pixel-based MV selection is performed in order to avoid blocking artifacts. For halo reduction, which requires both occlusion area detection and adequate ME, the problem is addressed in a fundamentally different way. Based on the notion of consistency between MV and intensity of adjacent pixels, a methodology using a sliding shape adaptive window (SAW), as used in [LeDinh, 2006], with local segmentation in three pixel groups, is proposed for removing the need for explicit C/U detection. For a sharp image interpolation, a generic adaptive MCI is introduced which considers the forward, backward interpolations and also the interpolated image position.

Section 3.2.2 of the paper presents in detail the proposed method. In Section 3.2.3, experimental results are given and discussed. Section 3.2.4 presents the final word on the techniques described in the paper.

3.2.2 Proposed frame rate conversion

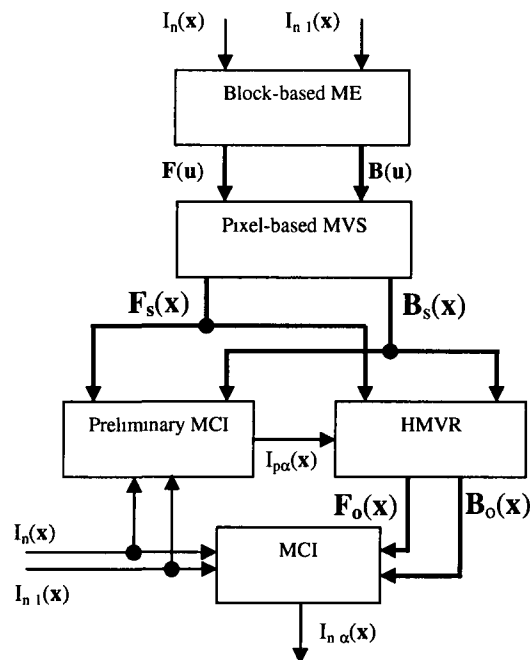


Figure 3.1 Proposed FRC

The system block diagram illustrated by Figure 3.1, accepts an existing progressive image $I_n(x)$ and its picture delayed version $I_{n-1}(x)$ as inputs and provides at the output an interpolated image $I_{n-\alpha}(x)$. Alpha, α , a fractional value, is the normalized distance between the existing

image $I_n(\mathbf{x})$ and the desired interpolated image $I_{n-\alpha}(\mathbf{x})$, meanwhile \mathbf{x} in bold character denotes the vector representing the current pixel coordinates in column and row

This section covers the details of the proposed FRC which is composed mainly of the block-based ME, pixel-based Motion Vector Selection, Halo consideration MV Refinement (HMVR) and MCI

A Block-based ME and Pixel-based MV Selection

The smoothness in MV field is an interesting property which has been exploited in several ways. Particularly, BMA for ME is an economical example of this technique which assumes pixels in a block should share the same MV. However, this assumption is no longer valid around moving object boundaries. Recently, a ME with variable block sizes [Chan *et al.*, 1990] and a MV interpolation for smaller block sizes [Dane et Nguyen, 2004] have been proposed in order to increase the MV field resolution. Pixel-based MV is desirable, however, the classical optical flow process [Aggarwal et Nandhakumar, 1988], [Tekalp, 1995] usually requiring several iterations makes it impractical for real-time applications. Particularly, for pixel-based FRC, Vatolin and Grishin proposed other designs [Vatolin et Grishin, 2006]. In the present FRC, the proposed pixel-based ME is composed of two steps. The first one is the well-known BMA followed by a second estimation or a selection of neighboring block MV for each pixel in the interpolated image. Moreover, in order to anticipate the image interpolation in occlusion regions, forward and backward MV estimations will be performed independently in BMA. Ideally, in non-occlusion regions, the forward and backward MVs are collinear and their sum is equal to zero, it would be equivalent to BDME's results. In occlusion regions, this property will no longer be valid, it could thus yield a method for occlusion detection.

Suppose forward motion vector (F) indicates the displacement of a given image block W of dimension $W \times W$ in image I_{n-1} , with respect to a matched block in image I_n . Backward motion vector (B) indicates the displacement of a given block in image I_n to a matched block in image I_{n-1} . If u represents the current block location, $u = \text{Integer}[x/W]$, the MV $F(u)$ and $B(u)$ minimize respectively the Mean of Absolute Differences Errors $\text{MADF}(u)$ and $\text{MADB}(u)$ for the exhaustive full search in a given search zone SZ .

$$MADF(\mathbf{u}) = \min_{\mathbf{F}} \frac{1}{n(\mathbf{R})} \sum_{\mathbf{y} \in \mathbf{R}} |I_{n-1}(\mathbf{u}\mathbf{W} + \mathbf{y}) - I_n(\mathbf{u}\mathbf{W} + \mathbf{y} + \mathbf{F})| \quad (3.1)$$

$$MADB(\mathbf{u}) = \min_{\mathbf{B}} \frac{1}{n(\mathbf{R})} \sum_{\mathbf{y} \in \mathbf{R}} |I_{n-1}(\mathbf{u}\mathbf{W} + \mathbf{y} + \mathbf{B}) - I_n(\mathbf{u}\mathbf{W} + \mathbf{y})| \quad (3.2)$$

in which \mathbf{R} denotes the current window region, $n(\mathbf{R})$ is the region size and \mathbf{y} are the coordinates of pixels within the window \mathbf{R} . The $MADF(\mathbf{u})$, $MADB(\mathbf{u})$ and the corresponding MVs can be associated to a small window \mathbf{W} centralized inside the reference region \mathbf{R} . Usually, the window \mathbf{W} size is $W \times W = 8 \times 8$. The implemented search zone \mathbf{SZ} is apt to detect displacements of ± 24 columns by ± 20 rows for standard definition TV (SDTV).

If the block-based MVs $\mathbf{F}(\mathbf{u})$ or $\mathbf{B}(\mathbf{u})$ are projected by following their linear trajectories, holes/covered areas (respectively, no/multiple MV) are an inevitable phenomenon produced in the interpolated image. In order to avoid this effect, as well as blocking artifacts, a MV Selection is used to provide for each pixel in the interpolated α -plane two dense MV fields corresponding to its own selected forward and backward MVs, $MV F_s(\mathbf{x})$ and $MV B_s(\mathbf{x})$, respectively. Let \mathbf{u} be the coordinates of the current block \mathbf{W} , $\mathbf{u} = \text{Integer}[\mathbf{x}/\mathbf{W}]$. Let \mathbf{p} be coordinates of a block in a square of neighboring blocks of the current one \mathbf{u} . The square size of neighboring blocks could be a function of α and the MV length, however, a fixed dimension 5×5 can be used for implementation.

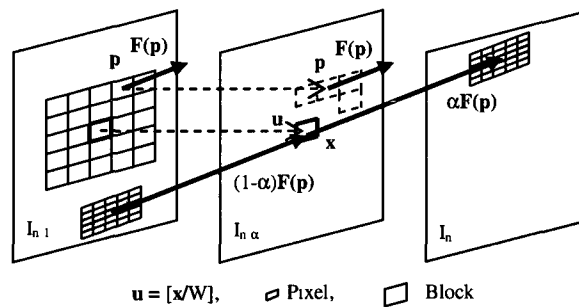


Figure 3.2 Illustration of forward MV used in MVS calculation

In a given α -plane, the pixel-based forward MV $F_s(\mathbf{x})$ is selected as one of the associated block-based Forward MV $\mathbf{F}(\mathbf{p})$ estimated in image I_{n-1} . As illustrated by Figure 3.2, the MV

$\mathbf{F}(\mathbf{p})$ is directly transferred from I_{n-1} to $I_{n-\alpha}$, i.e. it maintains the same spatial coordinates from frame to frame. $\mathbf{F}_S(\mathbf{x})$ minimizes the following Weighted Error Forward $WEF(\mathbf{x}, \mathbf{p})$

$$WEF(\mathbf{x}, \mathbf{p}) = a_f(\mathbf{p})[(1-c)MLEF(\mathbf{x}, \mathbf{p}) + cMADF(\mathbf{p})] + b_f(\mathbf{p}) \quad (3.3)$$

The obtained minimum error is denoted by

$$\begin{aligned} MWEF(\mathbf{x}) &= \min_{\mathbf{F}(\mathbf{p})} WEF(\mathbf{x}, \mathbf{p}) \\ &= WEF(\mathbf{x}, \mathbf{p}) \Big|_{\mathbf{F}(\mathbf{p})=\mathbf{F}_S(\mathbf{x})} \\ \mathbf{F}_S(\mathbf{x}) &= \arg \min WEF(\mathbf{x}, \mathbf{p}), \end{aligned} \quad (3.4)$$

In (3.3), $MADF(\mathbf{p})$ is defined previously as in (3.1), c is a fixed consistency factor for combining 2 errors: the block-based $MADF(\mathbf{p})$ and the pixel-based $MLEF(\mathbf{x}; \mathbf{p})$. Experimentally, c is set equal to $(1/4)$ to reflect the significant contribution of the pixel-based error over the block-based one. $MLEF(\mathbf{x}; \mathbf{p})$ denoting Mean Local absolute Error Forward is a mean bi-directional MCI error provided from a forward MV $\mathbf{F}(\mathbf{p})$ at the pixel \mathbf{x} in the α -plane

$$MLEF(\mathbf{x}, \mathbf{p}) = \frac{1}{n(\mathbf{K})} \sum_{\mathbf{k} \in \mathbf{K}} |I_n(\mathbf{x} + \mathbf{k} + \alpha \mathbf{F}(\mathbf{p})) - I_{n-1}(\mathbf{x} + \mathbf{k} - (1-\alpha) \mathbf{F}(\mathbf{p}))| \quad (3.5)$$

\mathbf{K} is a small local 5×5 window around the current pixel \mathbf{x} . The window size 5×5 yields reliable ME for SDTV resolution.

In (3.3), if $MLEF(\mathbf{x}; \mathbf{p})$ is the central mechanism for MVS, the factor $a_f(\mathbf{p})$ is generally equal to 1 and the penalty offset $b_f(\mathbf{p})$ is 0. However, when $MADF(\mathbf{p})$ is smaller than a threshold value and the component-wise absolute differences between the MV $\mathbf{F}(\mathbf{p})$ and its 5×5 average is bigger or equal to another experimentally set threshold, then the factor and the offset values are set to be respectively equal to 2 and 1. In practical terms, there is a penalty applied on a given block \mathbf{p} when its MAD is too small and its MV is relatively isolated.

Similarly, the same principle is applied independently for backward MV. In the α -plane, selected pixel-based $\mathbf{B}_S(\mathbf{x})$ can be obtained using the following equations

$$WEB(x, p) = a_b(p)[(1-c)MLEB(x, p) + cMADB(p)] + b_b(p) \quad (3.6)$$

$$\begin{aligned} MWEB(x) &= \min_{B(p)} WEB(x, p) \\ &= WEB(x, p) \Big|_{B(p)=B_s(x)} \end{aligned}$$

$$B_s(x) = \arg \min WEB(x, p) \quad (3.7)$$

$$MLEB(x, p) = \frac{1}{n(K)} \sum_{k \in K} |I_n(x+k-\alpha B(p)) - I_{n-1}(x+k+(1-\alpha)B(p))| \quad (3.8)$$

It should be noted that the estimated MV $F_s(x)$ and $B_s(x)$ are function of the α value. Moreover, as stated previously, in non-occlusion regions, $F_s(x) + B_s(x) = 0$. In occlusion regions, it can be seen that the ME model is flawed and since MVS is a combining technique of one-directional and bidirectional ME, it is thus difficult to verify if one of F_s or B_s , or neither is the true MV. The next section, concerning Halo MV Refinement, aims to address this obstacle. Furthermore, the short abbreviation MWE for Minimum Weighted local Error will be used when no forward or backward MV is to be specified.

As illustrated by Figure 3.3 in which the horizontal component of forward MV is displayed at the pixel resolution, the blocking artefact is strongly reduced after MVS.

For HDTV and higher resolution, the full search technique described above is applied in combination with a hierarchical approach to extend the footprint of the effective search window. MVS technique is then applied at different levels to reduce potential cumulative ME errors and to maintain good image and MV resolution.



(a)

(b)

backward MV \mathbf{B}_s , and the window size (MN) Let SF and SB be pixel-based versions of MADF and MADB, respectively SF(x) and SB(x) can be obtained from MADF(u) and MADB(u) by a WxW up-sampler, followed by suitable separable interpolation filters to match picture source pixel resolution Furthermore, let S1 and S2 be two threshold values The weighting $\gamma(\mathbf{x})$ can be defined more precisely as follows

If $\alpha \geq 1/2$, then (3 10)

$$\gamma(\mathbf{x}) = [\# \text{ pixels } (SB(\mathbf{x}) < S1) \text{ or } (S1 \leq SB(\mathbf{x}) \leq SF(\mathbf{x}) + S2)] / (MN),$$

If $\alpha < 1/2$, then

$$\gamma(\mathbf{x}) = 1 - [\# \text{ pixels } (SF(\mathbf{x}) < S1) \text{ or } (S1 \leq SF(\mathbf{x}) \leq SB(\mathbf{x}) + S2)] / (MN),$$

In other words, the weighting factor $\gamma(\mathbf{x})$ is not based on an individual MADB result, but on a context-based decision which is more adaptive, and in turn, more reliable

Note that for the preliminary image $I_{p,\alpha}(\mathbf{x})$, the interpolations required in (3 9) can be performed using simple linear filtering

b) Halo consideration Motion Vector Refinement HMVR

The HMVR concept is based on an IxJ sized sliding window illustrated by Figure 3 4 in which there are 3 groups of pixels in the vicinity of an occlusion area

In Figure 3 4a when the considered pixel is in a foreground region, Group 1 consists heuristically of a local group of pixels with small weighted local error MWE and similar intensity level and similar MV values Group 2 consists of a local group of pixels with large MWE and different intensity level but similar MV values to that of the central pixel Group 3 consists of a local group of pixels with small MWE and different intensity levels and different MV values than that of the central pixel Commonly, a large error is not a reliable indicator of a halo region However, by combining this information with the local segmentation of vectors and pixel intensities, the identification and grouping of the different regions allow for better halo separation and makes possible robust MV refinement as described below

Figure 3 4b illustrates the case where the central pixel is in the occlusion area There are, again, 3 groups of pixels of interest Group 1 consists of a local group of pixels with small

MWE and similar MV value but different intensity level from the central pixel of the window Group 2 consists of a local group of pixels with large MWE and similar intensity levels and similar MV values to that of the central pixel of Figure 3 4b Group 3 consists of a local group of pixels with small MWE and different MV values but with similar intensity levels to that of the window's central pixel

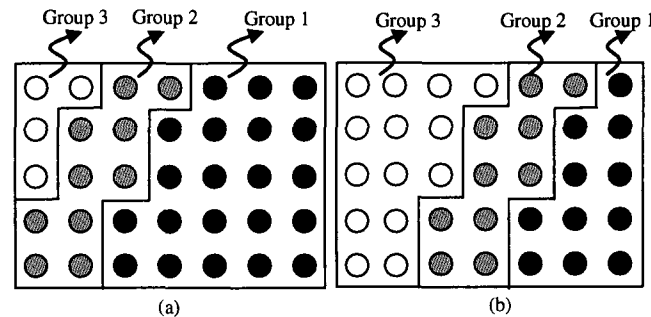


Figure 3 4 Windows contains Foreground (black pixel) relatively moving to the left, Occluded Background (grey), Non-occluded Background (white) (a) Considered (central) pixel in foreground region, (b) Considered pixel in occlusion area

Figure 3 4 illustrating particular C/U situations, provides a framework to help model a MV correction method As an example, for the case of Figure 3 4a, MV at the considered pixel can be unchanged or can be lightly changed for smoothness purpose by the mean MV of the pixels in the same group For the case of Figure 3 4b, MV at the considered pixel has to be changed by the mean MV of the pixels in the window with similar intensities (within a threshold value) but different MV yielding small MWE

The group of pixels having a set of given properties (such as similar intensities and similar MV to that of the central pixel) can be seen as a Shape Adaptive Window (SAW) This is a locally segmented rectangular sliding window defined by weighting elements 1 or 0 “1” indicates pixels in a window region with similar properties to the considered pixel, “0” corresponds to the negative cases The component-wise Mean MV provided by such SAW can be interpreted as the result of a generic version of the well-known Mean Shift procedure or filtering in the literature [Fukunaga, 1975], [Cheng, 1995] Intensity-based SAW filtering has been utilized for noise or coding artifact reduction [LeDinh, 2006] in the past

The window size, I by J, should be large enough to obtain a sufficient number of pixels representing each of the groups when the current pixel is located in a suspected halo region. Moreover, I and J should be small enough to avoid irrelevant information which can falsify the MV estimation. In our implementation, the window is asymmetric as J is equal to 5 lines for hardware-friendly implementation and I is equal to 11 columns so that it is bigger than the window W 's length, $W=8$.

Aside from the missing information on MWE, Figure 3.4 also provides some indication for MV refinement detection even for the case of Figure 3.4b in which the considered pixel is situated in the occluded group with similar intensity and similar (but false) MV. The indication can be N_R , the number of pixels that are similar in intensity and MV value to the considered central pixel. The idea is based on the following observation: if the window size is large enough, the number of occluded pixels is relatively small.

Let N_D be the number of pixels of similar intensity but different MV values to that of the central pixel. Let N_{RD} be the number of pixels with similar intensities as the central pixel in the window. As a result, $N_{RD} = N_R + N_D$. N_{RD} is smaller than the window size.

Figure 3.5 illustrates the Decision Map (DM) for MV correction. Depending on the current pixel position, N_{RD} is a varying number whereas $I \times J$ is a fixed window size value. In order to get a stable and coherent decision boundary, it is important to normalize the DM's horizontal and vertical axis respectively with the maximum numbers IJ and $N_{RD}(x)$. As such, for a given $N_{RD}(x)$, the normalized ratios for the current decision are determined by

$$\begin{aligned} a_R(x) &= N_R(x) / (IJ) \\ b_R(x) &= N_R(x) / N_{RD}(x) \end{aligned} \quad (3.11)$$

If the resulting point (a_R, b_R) falls in the zone Z1, as illustrated, the current MV will be substituted by the mean MV of pixels with similar intensities but different MV. In the zone Z3, in the right hand high corner of DM, the current MV can be changed by the mean MV of pixels with similar intensities and similar MV. Otherwise, in Z2, the current MV will be preserved. Therefore, the proposed MV correction process does not require explicit detection of C/U regions and can apply for all pixels in the to-be-interpolated image without any distinction.

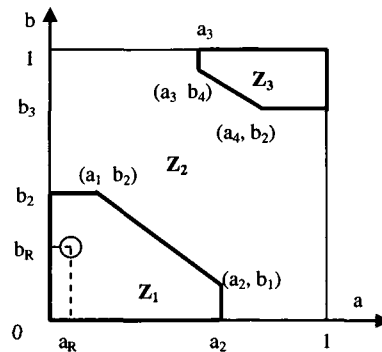


Figure 3.5 Decision Map (a, b) Z_1 Select different MV, Z_2 Keep the current MV, Z_3 Select similar MV $a_R = N_R/(IJ)$, $b_R = N_R/N_{RD}$

Thus, as illustrated by Figure 3.4b, the MV in Group 2 will be substituted by the mean value of motion vectors of Group 3. In Figure 3.4a, the MV in the Group 1 will not be corrected.

A pseudo code is given to summarize the HMVR' mechanism below.

Let $q \in$ Sliding Window of dimensions I, J centered at x ,

Let $V_s(x)$ MV input, $V_s(x) = F_s(x)$ or $B_s(x)$,

Let $V_I(x)$ HMVR output,

Let $I_p \alpha(x)$ Preliminary MCI,

Let $LI_p \alpha(x)$ Low-pass filtered version of $I_p \alpha(x)$

Shape Adaptive Window Local Segmentation by suitable thresholds,

$$\omega_R(q, x) = 1 \text{ if } V_s(q) \cong V_s(x) \text{ and } LI_p \alpha(q) \cong LI_p \alpha(x)$$

$$\omega_R(q, x) = 0 \text{ if else}$$

$$N_R(x) = (\text{Number of } \omega_R(q, x) = 1)$$

$$\omega_D(q, x) = 1 \text{ if } V_s(q) \neq V_s(x) \text{ and } LI_p \alpha(q) \cong LI_p \alpha(x)$$

$$\omega_D(q, x) = 0 \text{ if else}$$

$$N_D(x) = (\text{Number of } \omega_D(q, x) = 1)$$

Number of pixels of similar intensities,

$$N_{RD}(x) = N_R(x) + N_D(x)$$

Normalized Ratio for decision and correction,

$$a_R(x) = N_R(x)/(IJ)$$

$$b_R(\mathbf{x}) = N_R(\mathbf{x}) / N_{RD}(\mathbf{x})$$

Decision and Correction,

$$\text{If } (a_R, b_R) \in \mathbf{Z1}$$

$$V_I(\mathbf{x}) = \text{SAW Mean } (V_s(\mathbf{q})) = \sum_q (V_s(\mathbf{q}) \omega_D(\mathbf{q}, \mathbf{x})) / N_D(\mathbf{x})$$

$$\text{If } (a_R, b_R) \in \mathbf{Z3}$$

$$V_I(\mathbf{x}) = \text{SAW Mean } (V_s(\mathbf{q})) = \sum_q (V_s(\mathbf{q}) \omega_R(\mathbf{q}, \mathbf{x})) / N_R(\mathbf{x})$$

$$\text{If else, } V_I(\mathbf{x}) = V_s(\mathbf{x}), \quad (3\ 12)$$

The proposed HMVR is applied separately to the forward and backward MV, $\mathbf{F}_s(\mathbf{x})$ and $\mathbf{B}_s(\mathbf{x})$. Let the resulting MV be $\mathbf{F}_1(\mathbf{x})$ and $\mathbf{B}_1(\mathbf{x})$.

Since HMVR is composed of various pixel-based hard decisions, a MV Blending and Isolated MV Filtering are proposed for smoothing decision transitions. These latter blocks are necessary to minimize the remaining artifacts left when halo is properly reduced.

MV Blending is a soft mixing between the refined MV and its corresponding selected MV input from MVS. More accurately, the forward MV $\mathbf{F}_1(\mathbf{x})$ will be combined with the selected MV $\mathbf{F}_s(\mathbf{x})$ using a pixel-based weighting factor $fe(\mathbf{x})$. Let $\mathbf{F}_2(\mathbf{x})$ be the blending result output given as follows:

$$\mathbf{F}_2(\mathbf{x}) = \mathbf{F}_s(\mathbf{x}) + [\mathbf{F}_1(\mathbf{x}) - \mathbf{F}_s(\mathbf{x})] fe(\mathbf{x}) \quad (3\ 13)$$

In (13) the weighting factor $fe(\mathbf{x})$ can be given by the following equations in which $fc(\mathbf{x})$ is the comparator output for MWEF(\mathbf{x}) and $lp(\mathbf{x})$ is the impulse response of the low-pass filter:

$$\begin{aligned} fe(\mathbf{x}) &= fc(\mathbf{x}) * lp(\mathbf{x}) \\ fc(\mathbf{x}) &= \begin{cases} 1, & \text{if } MWEF(\mathbf{x}) \geq \text{Threshold} = 2 \\ 0, & \text{if else} \end{cases} \\ lp(\mathbf{x}) &= \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} / 16 \end{aligned} \quad (3\ 14)$$

In a similar manner, $\mathbf{B}_2(\mathbf{x})$, the blended version of $\mathbf{B}_1(\mathbf{x})$, is given by:

$$\mathbf{B}_2(\mathbf{x}) = \mathbf{B}_s(\mathbf{x}) + [\mathbf{B}_1(\mathbf{x}) - \mathbf{B}_s(\mathbf{x})] be(\mathbf{x}) \quad (3\ 15)$$

where $be(\mathbf{x})$ is a pixel-based weighting factor provided by backward estimation error of MWEB(\mathbf{x})

$$be(\mathbf{x}) = bc(\mathbf{x}) * lp(\mathbf{x})$$

$$bc(\mathbf{x}) = \begin{cases} 1, & \text{if } MWEB(\mathbf{x}) \geq \text{Threshold} = 2 \\ 0, & \text{if } \text{else} \end{cases} \quad (3.16)$$

In non-occlusion regions, the Mean Weighted local Error MWE resulting from pixel-based bidirectional and block-based one-directional ME is relatively small. A low value (< 2) threshold given as in (14) and (16) can justify the soft mixing decision in (13) and (15) by considering the following

By keeping the threshold value low, the selection of the corrected MV, is overly favored. However, when used in combination with the HMVR, which is already constrained by a number of conditions and generates these corrected MV, the over-detection of MV Blending block is masked by the encapsulation of HMVR's regional properties and thus, becomes a non-issue. Of course, more elaborate decisions, for example by combining the relation $\mathbf{F} + \mathbf{B} = \mathbf{0}$ in non-occlusion regions, could be possible. However, the problem of isolated MV remains. Often, a case identified as Isolated MV and not properly addressed will create isolated and rather distracting *spikes* of pixel intensities.

The present Isolated MV Filtering is an adaptive median filter. If the blended MV $F_2(\mathbf{x})$ is isolated relatively to the 3x3 neighboring MV $F_2(\mathbf{q})$, then the filter output $F_0(\mathbf{x})$ is given by the 9-points median forward MV $F_2(\mathbf{q})$. Precisely, the component-wise MV filtering can be described as follows

Let $F_{2c}(\mathbf{q})$ and $F_{2r}(\mathbf{q})$ be the horizontal and vertical components of $F_2(\mathbf{q})$ in the window. The horizontal and vertical components of $F_0(\mathbf{x})$ filtered output MV are determined as

$$N_c(\mathbf{x}) = \text{Number } F(\mathbf{q}) \mid |F_{2c}(\mathbf{q}) - F_{2c}(\mathbf{x})| < Th = 3$$

$$F_{oc}(\mathbf{x}) = \begin{cases} \text{Median}(F_{2c}(\mathbf{q})), & \text{if } N_c(\mathbf{x}) \leq NTh = 3 \\ F_{2c}(\mathbf{x}), & \text{else} \end{cases} \quad (3.17)$$

$$N_r(\mathbf{x}) = \text{Number } F(\mathbf{q}) \mid |F_{2r}(\mathbf{q}) - F_{2r}(\mathbf{x})| < Th = 3$$

$$F_{or}(\mathbf{x}) = \begin{cases} \text{Median}(F_{2r}(\mathbf{q})) & \text{if } N_r(\mathbf{x}) \leq NTh = 3 \\ F_{2r}(\mathbf{x}), & \text{else} \end{cases}$$

Similar filtering is also applied for providing backward MV $B_o(x)$

The resulting forward and backward $F_o(x)$ and $B_o(x)$ are now utilized for MCI

C MCI

If the first proposed MCI is only approximated for analysis, the second MCI shall be more elaborate to further reduce artefacts. A possibility is to imply the α position. If the interpolated image is near from the existing image I_n , it will be reasonable to use only I_n . Inversely, at the other end, it will be I_{n-1} . This consideration will be useful, as an example, for a 5-times conversion from 24 to 120 Hz. The following interpolation is proposed for MC interpolated image $I_{n-\alpha}(x)$

$$\begin{aligned} & \text{If } 1 > \alpha \geq \alpha_H, \\ I_{n-\alpha}(x) &= (1-\gamma) I_{n-1}(x - (1-\alpha) F_o(x)) + \gamma I_{n-1}(x + (1-\alpha) B_o(x)) \end{aligned} \quad (3.18)$$

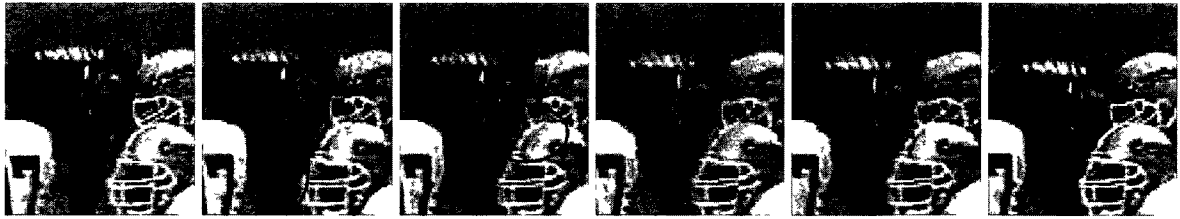
$$\begin{aligned} & \text{If } \alpha_L \geq \alpha > 0, \\ I_{n-\alpha}(x) &= (1-\gamma) I_n(x + \alpha F_o(x)) + \gamma I_n(x - \alpha B_o(x)) \end{aligned} \quad (3.19)$$

$$\begin{aligned} & \text{If } \alpha_H > \alpha > \alpha_L, \\ I_{n-\alpha}(x) &= (1-\gamma) I_n(x + \alpha F_o(x)) + \gamma I_{n-1}(x + (1-\alpha) B_o(x)) \end{aligned} \quad (3.20)$$

In (18)-(20), the local statistic context-based weighting factor γ , function of both α and x , is defined as in (10). The $F_o(x)$ and $B_o(x)$ are also function of α as described in the HMVR subsection. For $\alpha = 1/2$, (18) and (20) are identical. Finally, α_H and α_L are fixed values and $\alpha_H + \alpha_L = 1$.

The proposed MCI thus provides sharp interpolated images and yields a smoothed temporal transition between an existing image and its neighboring interpolated images.

Figure 3.6 illustrates at the same time the proposed MVS, HMVR and MCI techniques. It represents a 5-times interpolation between two existing images, frame 93 and 94, i.e. $\alpha = 1, 0.8, 0.6, 0.4, 0.2$ and 0 . The first and second rows represent the results without and with MVS and HMVR, respectively. Results from the latter are noticeably sharper.



(a)



(b)

Figure 3.6 *Football* sequence – From left to right $\alpha = 1, 0.8, 0.6, 0.4, 0.2, 0$

(a) Interpolation without both MVS and HMVR (b) Interpolation with MVS and HMVR

3.2.3 Experimental results

In this section, various video sequences have been tested to evaluate the performance of the proposed method. To compare the performance, every frame from the test sequences is skipped and re-interpolated with two original adjacent frames by the traditional bidirectional method (BDME), the proposed model without halo correction, the whole proposed model as well as the pixel-based FRC model as in [Vatolin et Grishin, 2006]. The interpolated frames are then compared with the corresponding original skipped frames.

The traditional bidirectional (BDME) method is implemented with a 16x16 block size, which gives better PSNR performance than 8x8 block size in our implementation. The pixel-based FRC interpolation related to [Vatolin et Grishin, 2006] is performed with “High Quality” option.

A Objective Evaluation

The objective evaluation is based on PSNR defined as

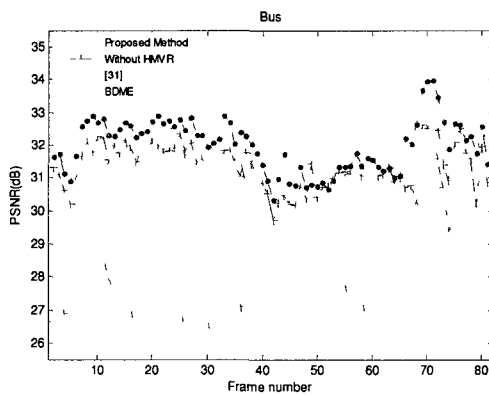
$$PSNR = 10 \log_{10} \left(\frac{255^2}{\frac{1}{ColorImageSize} \sum_x (I_n(x) - \hat{I}_n(x))^2} \right) \quad (3.21)$$

in which $I_n(x)$ and $\hat{I}_n(x)$ are respectively the original and the interpolated color images

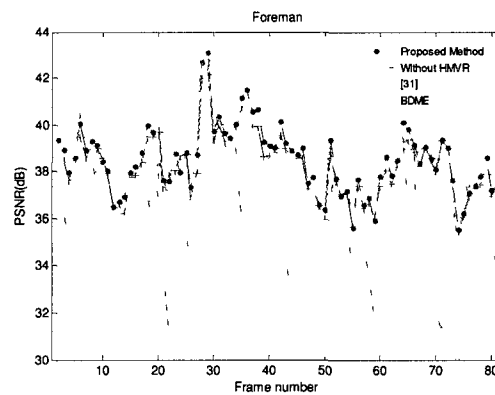
Figure 3 7 illustrates the frame-by-frame PSNR evolution of the four methods respectively for *Bus*, *Foreman*, *Cheerleaders* and *Calendar Mobile* sequences It can be seen that the proposed method yields a superior performance consistency throughout Tableau 3 1 represents a PSNR comparison for six video sequences The proposed method outperforms the other considered existing methods Moreover, the proposed algorithm is based on feedforward technique suitable for pipelined implementation and aimed for real-time processing using contemporary IC technology

Table 3 1
PSNR Performance Comparisons among Four Frame Interpolation Methods
for Six Video Sequences

| | Proposed Method | Without HMVR | [Vatolin et Grishin, 2006] | BDME |
|----------------------|-----------------|--------------|----------------------------|-------|
| <i>Bus</i> | 32.00 | 31 36 | 31 57 | 27 44 |
| <i>Foreman</i> | 38.51 | 38 24 | 37 75 | 35 00 |
| <i>Garden Flower</i> | 33.29 | 33 20 | 32 24 | 30 81 |
| <i>Mobile</i> | 32.93 | 32 75 | 31 67 | 26 09 |
| <i>Ferris wheel</i> | 41.23 | 40 71 | 37 45 | 40 05 |
| <i>Cheerleaders</i> | 29.82 | 29 06 | 29 37 | 27 71 |



(a)



(b)

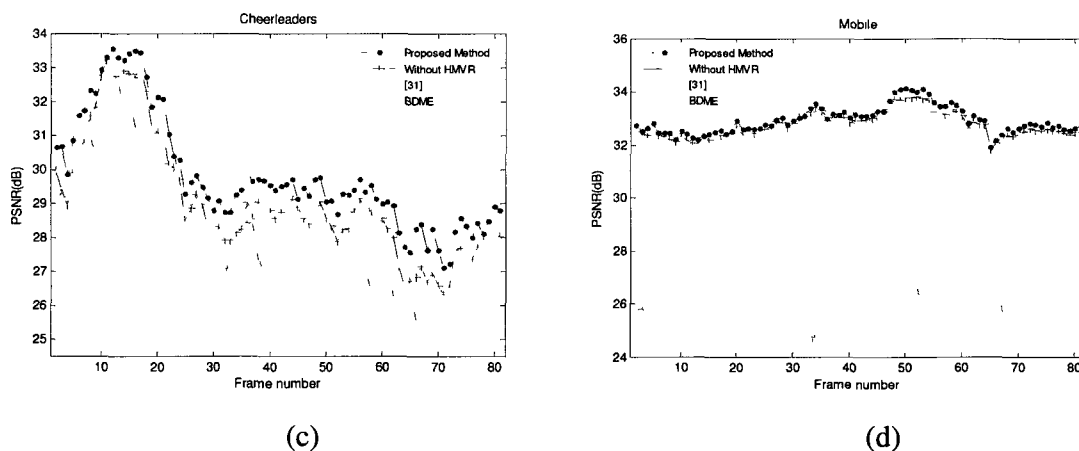


Figure 3.7 Comparison of PSNR performance

(a) *Bus*, (b) *Foreman*, (c) *Cheerleaders*, (d) *Calendar Mobile*

B Subjective Evaluation

Since PSNR represents an average or a global fidelity measure between the interpolated image and the original one for the whole image, it cannot accurately represent the image quality perceived by the human visual system. Subjective image quality is also important to evaluate in any application that targets a real implementation.

Figure 3.8 and figure 3.9 show the original images *Foreman* and *Bus* together with the interpolated results of the considered methods. Figure 3.8 represents cropped images in a window and their associated SSIM values [Wang *et al.*, 2004]. For the image *Bus* in figure 3.9, halo artefacts and/or deformed structures can be observed in the resulting images from existing methods. More interpolated results can be viewed at <http://www.gel.usherbrooke.ca/tran/>

The interpolated frame yielded by the proposed model exhibits the best subjective visual quality.

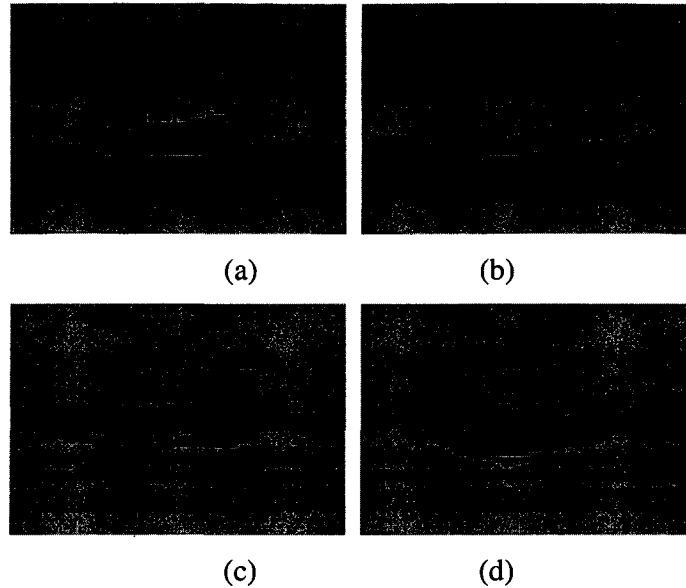


Figure 3.8 Cropped images – *Foreman* sequence – frame 57 (a) Original, (b) [Vatolin et Grishin, 2006], SSIM = 0.93, (c) without both MVS and HMVR, SSIM = 0.91, (d) Proposed Method, SSIM = 0.98

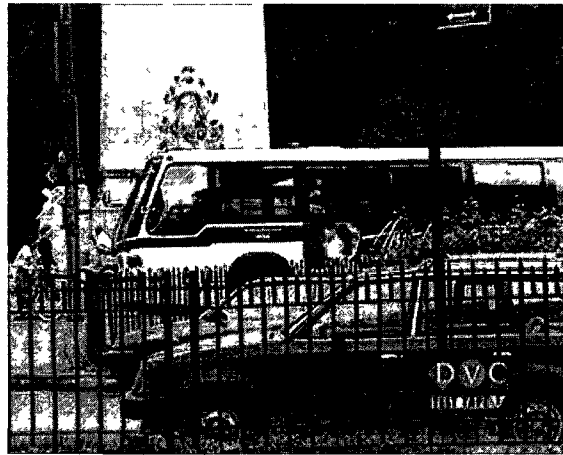
3.2.4 Conclusion

In this paper, we propose a new FRC method based on MV Selection, Halo consideration and MV Refinement. Combined with the classical BMA, MV Selection is an effective way to get dense MV field in an interpolated plan without holes or overlapped areas. For Halo reduction, a three-group model in a sliding window is introduced. Shape Adaptive Window defined at the same time by MV and preliminary interpolated intensity yields a possibility for false MV correction without explicit occlusion detection. MCI for many times interpolation is extended using interpolated plane position information and local statistic context-based weighting factor, (γ)

An immediate extension of the work can be foreseen by combining it with a hierarchical approach for HD or higher resolution.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their useful comments.



(a)



(b)



(c)



(d)



(e)

Figure 3 9 *Bus* sequence (a) Original frame 33, (b) [Vatolin et Grishin, 2006], (c) Without both MVS and HMVR, (d) Bi-directional (BDME), (e) Proposed Method

3.3 Traitements supplémentaires

Cette section porte sur certains traitements supplémentaires qui ne sont pas présentés dans la publication de la recherche [Tran et LeDinh, 2011] à cause du nombre de page limité. Ces traitements supplémentaires ne sont pas nécessairement génériques, par contre, dans certaines situations, ils complètent bien les techniques proposées à la section précédente. On abordera spécifiquement les deux problèmes suivants : a) réduction supplémentaire de l'effet de halo dans un arrière-plan périodique et quasi statique et b) réduction de l'effet de scintillement dans le domaine de l'image.

3.3.1 Réduction de l'effet de halo dans un arrière-plan périodique et quasi statique

Les bons vecteurs de mouvement dans la zone d'occlusion ne sont pas souvent trouvés. Si l'arrière-plan est plat ou flou, le résultat ne peut pas démontrer clairement l'effet de halo ou l'effet de bloc en cas de vecteur de mouvement incorrect. Pourtant, l'artefact est très visible dans un arrière-plan périodique quasi stationnaire. La figure 3.10 illustre un exemple de cet artefact qui se manifeste autour du casque du joueur.



Figure 3.10 Artefact de halo dans l'arrière-plan périodique

Cet artefact de halo n'a pas été détecté selon les techniques proposées à la section précédente, car l'intensité moyenne du chandail de l'arbitre est similaire à la partie luisante du casque. Nous proposons une solution pour réduire cet artefact selon le schéma de la figure 3.11 suivante.

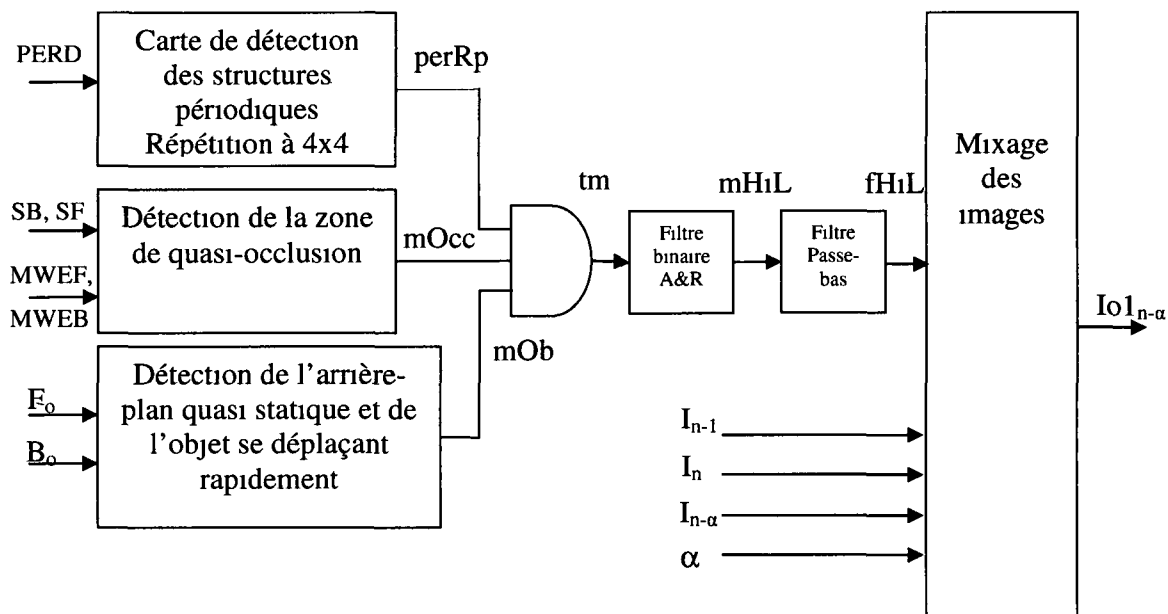


Figure 3 11 Schéma de la réduction de l'effet de halo dans un arrière-plan périodique et quasi statique

Selon le schéma, le signal binaire mHiL est la détection de l'effet de halo dans un arrière-plan périodique et quasi statique

mHiL est égal à 1 si

- le pixel considéré est détecté comme un pixel dans une zone périodique $perRp(x) = 1$
- et il est dans une zone d'occlusion $mOcc(x) = 1$
- et les pixels quasi immobiles ainsi que les pixels déplacés rapidement se trouvent dans une région avoisinante de pixel considéré $mOb(x) = 1$

La détection rudimentaire tm de mHiL est produite donc par les portes logiques « et »

$$tm(x) = perRp(x) \text{ et } mOcc(x) \text{ et } mOb(x) \quad (3\ 22)$$

Par la suite, tm est filtrée par un filtre « d'ajout-enlèvement », A&R, [LeDinh, 2003] de taille 3x3 afin d'ajouter les pixels manqués ou d'enlever les pixels écartés selon leurs voisins. Ensuite, un filtre passe-bas est appliqué pour lisser le mixage selon le filtrage suivant (3 23)

$$fHiL(x) = lp(x) * mHiL(x) \quad (3\ 23)$$

$$lp(\mathbf{x}) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} / 16 \quad (3.24)$$

À la figure 3.11 PERD est un signal binaire obtenu du bloc de détection de la périodicité décrit dans [Rossignol *et al.*, 2009, (USP0161763A1)] Voici l'essentiel de la détection de périodicité l'image à l'entrée est soumise à deux analyses FFT (Fast Fourier Transform) basées bloc selon les deux directions horizontale et verticale À l'aide de seuillage approprié, ces analyses procurent à chacune des analyses une détection possible de l'existence des composantes périodiques La détection composée, via une simple porte logique « ou », est ensuite filtrée par les filtres binaires « d'ajout-enlèvement » [LeDinh, 2003] dans le but de consolider la détection selon le contexte local pour former le signal PERD

Le signal PERD est répété dans chaque bloc de (4x4), comme illustre la figure 3.11, pour donner la résolution désirée compatible avec les autres signaux

L'effet de halo peut se produire à l'arrière-plan dans une zone d'occlusion lorsqu'à l'avant-plan un objet se déplace rapidement L'erreur d'appariement dans cette zone n'est souvent pas petite mOcc est obtenue par une combinaison de SF, SB, MWEF et MWEB comme suit

$$mOcc(\mathbf{x}) = ((SF(\mathbf{x}) > Th1) \text{ ou } (SB(\mathbf{x}) > Th1)) \text{ et } ((MWEF(\mathbf{x}) > Th2) \text{ ou } (MWEB(\mathbf{x}) > Th2)) \quad (3.25)$$

De plus, le signal mHiL(x) requiert aussi une détection combinée mOb(x) d'un arrière-plan quasi statique et d'un objet se déplaçant rapidement Le pseudo-code suivant est suggéré pour cette détection

1 Balayer pour toute l'image

$$\text{Soit } M(\mathbf{x}) = \max(|F_{oc}(\mathbf{x})|, |F_{or}(\mathbf{x})|, |B_{oc}(\mathbf{x})|, |B_{or}(\mathbf{x})|)$$

2 Balayer pour toute l'image

$$Cp(\mathbf{x}) = \text{compter pour } M(\mathbf{t}) < Sp \text{ dans une fenêtre } 7 \times 9 \text{ autour de } \mathbf{x}$$

$$Cg(\mathbf{x}) = \text{compter pour } M(\mathbf{t}) > Sg \text{ dans une fenêtre } 3 \times 5 \text{ autour de } \mathbf{x}$$

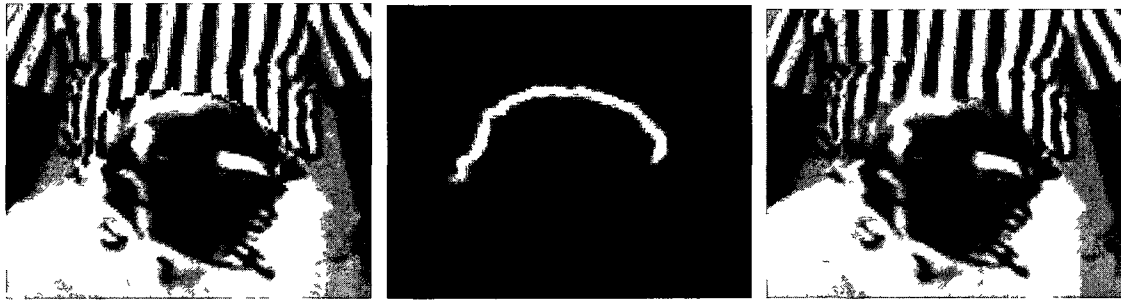
3 $mOb(\mathbf{x}) = (Cp(\mathbf{x}) > smMV) \text{ et } (Cg(\mathbf{x}) > gtMV) \quad (3.26)$

En pratique, Th1 est choisi à 10 et Th2 est choisi à 6, Sp est de 3 et Sg est de 5 Tous ont les valeurs de 8 bits

L'image à la sortie du mixeur est calculée par la formule suivante

$$Io_{1_{n-\alpha}}(\mathbf{x}) = (((I_{n-1}(\mathbf{x}) - I_n(\mathbf{x})) \alpha + I_n(\mathbf{x})) - I_{n-\alpha}(\mathbf{x})) f_{HiL}(\mathbf{x}) + I_{n-\alpha}(\mathbf{x}) \quad (3.27)$$

La figure 3.12 démontre que le halo est vraiment réduit autour du casque du joueur de football



a) Avant la réduction

b) La carte de la réduction

c) Après la réduction

Figure 3.12 Réduction de l'effet de halo dans l'arrière-plan périodique quasi statique

3.3.2 Post-traitement pour scintillement

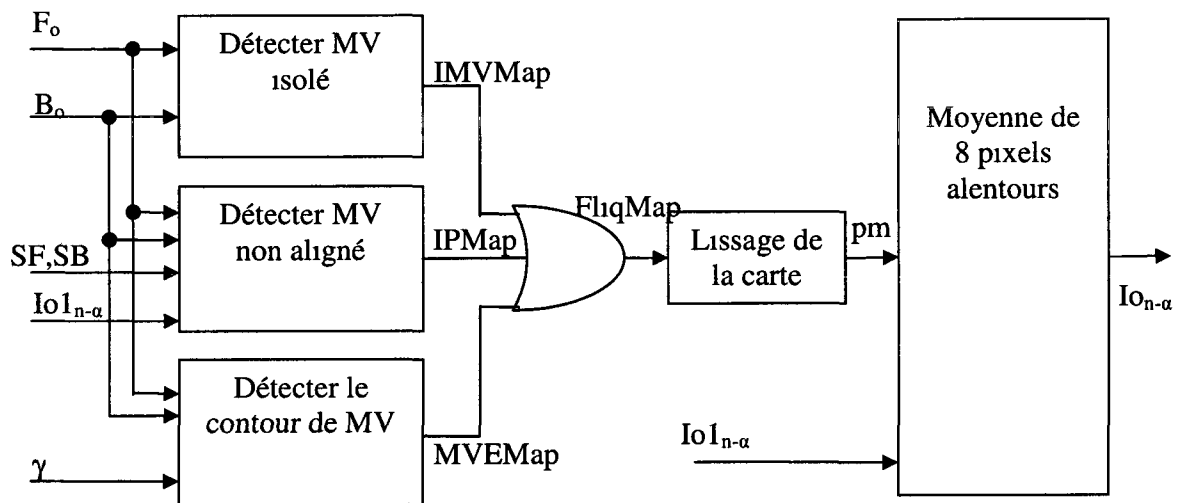


Figure 3.13 Schéma bloc du post-traitement pour scintillement

L'effet de scintillement ou de clignotement est dû principalement aux vecteurs de mouvement isolés ou des pixels interpolés erronés et isolés dans les régions d'occlusion, d'une part. D'autre part, c'est possiblement dû à l'imprécision d'estimation de mouvement à la bordure

des régions de différents vecteurs de mouvement dans un champ de mouvement Dans cette section, une solution proposée pour réduire cet artefact est illustrée à la figure 3 13 suivante La solution comprend principalement trois détections celle des vecteurs de mouvement isolés (pour IMVMap), celle des vecteurs de mouvement non-alignés (pour IPMap) et celle du contour des vecteurs de mouvement (pour MVEMap)

Le vecteur de mouvement isolé est défini à la position de pixel au cours de laquelle l'une des 4 composantes F_{oc} , F_{or} , B_{oc} et B_{or} n'a pas plus de 2 valeurs similaires en comparaison d'un seuil de ± 1 dans une fenêtre glissante locale de 3×3 Le pseudo-code suivant présente cette détection des vecteurs de mouvement isolés

Balayer p dans la fenêtre 3×3 où x est pixel au centre

$$NFc(x) = \text{compter pour } Abs(F_c(p) - F_c(x)) < ThV,$$

$$NFr(x) = \text{compter pour } Abs(F_r(p) - F_r(x)) < ThV$$

$$NBc(x) = \text{compter pour } Abs(B_c(p) - B_c(x)) < ThV$$

$$NBr(x) = \text{compter pour } Abs(B_r(p) - B_r(x)) < ThV$$

$$IMVMap(x) = NFc(x) \leq 3 \text{ ou } NFr(x) \leq 3 \text{ ou } NBc(x) \leq 3 \text{ ou } NBr(x) \leq 3 \quad (3.28)$$

L'effet causé par quelques MVs isolés est pourtant difficile à reconnaître lorsque la vidéo se déroule en temps réel

Dans la zone de non -occlusion, les vecteurs de mouvement $F_o(x)$ et $B_o(x)$ sont généralement égaux, c'est-à-dire $F_o(x) + B_o(x) = 0$ Si $F_o(x) + B_o(x) = d$, et $|d|$ est assez grand par rapport à $|F_o(x)|$ ou à $|B_o(x)|$, alors le pixel de coordonnées x se trouve peut-être dans une zone d'occlusion dans laquelle l'estimation de mouvement de x ne soit pas correcte SF, SB sont utilisées comme indicateurs pour situer plus ou moins une zone d'occlusion C'est bien de noter que même si cette détection d'occlusion n'est pas très précises, la correction, dont on va discuter après, ne rend floue que la zone détectée et cette zone est souvent petite par rapport à l'image entière Le pseudo-code suivant résume la détection des pixels isolés de grandes erreurs et des vecteurs de mouvement non alignés illustrés à la figure 3 12

, *Détection des pixels isolés de grandes erreurs et de vecteurs de mouvement non alignés*

Soient $SF(\mathbf{x})$ et $SB(\mathbf{x})$ sont les versions interpolées basées pixel de $MADB(\mathbf{u})$ et $MADB(\mathbf{u})$ respectivement

$$BadErr(\mathbf{x}) = (SF(\mathbf{x}) > ThErr) \text{ ou } (SB(\mathbf{x}) > ThErr)$$

$$IPX(\mathbf{x}) = Abs(I_c(\mathbf{x}) - (Moyenne \text{ des } 8 \text{ Intensités avoisinantes})) > ThI$$

$$M(\mathbf{x}) = \max (Abs(F_{oc}(\mathbf{x})), Abs(F_{or}(\mathbf{x})), Abs(B_{oc}(\mathbf{x})), Abs(B_{or}(\mathbf{x})))$$

$$ThAlign(x) = LUT(M(x))$$

$$Unalign(x) = (Abs(F_{oc}(x) + B_{oc}(x)) > ThAlign(x)) \text{ ou } (Abs(F_{or}(x) + B_{or}(x)) > ThAlign(x))$$

$$IPMap(\mathbf{x}) = BadErr(\mathbf{x}) \text{ et } IPX(\mathbf{x}) \text{ et } Unalign(\mathbf{x}) \quad (3 \ 29)$$

L'imprécision de l'estimation de mouvement dans le temps aux bordures des régions de différents vecteurs de mouvement dans un champ de mouvement peut constituer une source de clignotement

Le pseudo-code suivant décrit la détection de contour des vecteurs de mouvement utilisés pour faire l'interpolation d'image selon γ

, *Contour des vecteurs de mouvement*

$$\text{Soit } \mathbf{F}_o(\mathbf{x}) = (F_{oc}(\mathbf{x}), F_{or}(\mathbf{x}))$$

$$\text{Soit } SobelModFc(\mathbf{x}) \text{ est Somme des Modules des 4 compas de Sobel de } F_c(\mathbf{x})$$

$$\text{Soit } SobelModFr(\mathbf{x}) \text{ est Somme des Modules des 4 compas de Sobel de } F_r(\mathbf{x})$$

$$\text{Soit } \mathbf{B}_o(\mathbf{x}) = (B_{oc}(\mathbf{x}), B_{or}(\mathbf{x}))$$

$$\text{Soit } SobelModBc(\mathbf{x}) \text{ est Somme des Modules des 4 compas de Sobel de } B_c(\mathbf{x})$$

$$\text{Soit } SobelModBr(\mathbf{x}) \text{ est Somme des Modules des 4 compas de Sobel de } B_r(\mathbf{x})$$

$$FMap(\mathbf{x}) = SobelModFc(\mathbf{x}) > ThM \text{ ou } SobelModFr(\mathbf{x}) > ThM$$

$$BMap(\mathbf{x}) = SobelModBc(\mathbf{x}) > ThM \text{ ou } SobelModBr(\mathbf{x}) > ThM$$

$$MVEMap(\mathbf{x}) = (FMap(\mathbf{x})=1) \text{ et } (\gamma(\mathbf{x}) \leq 3/4) \text{ ou } ((BMap(\mathbf{x})=1) \text{ et } (\gamma(\mathbf{x}) \geq 1/4)) \quad (3 \ 30)$$

La carte de détection du post-traitement $FliqMap$ est la combinaison de trois cartes détectées de $IMVMap$, $IPMap$ et $MVEMap$ respectivement décrites dans (3 28), (3 29) et (3 30)

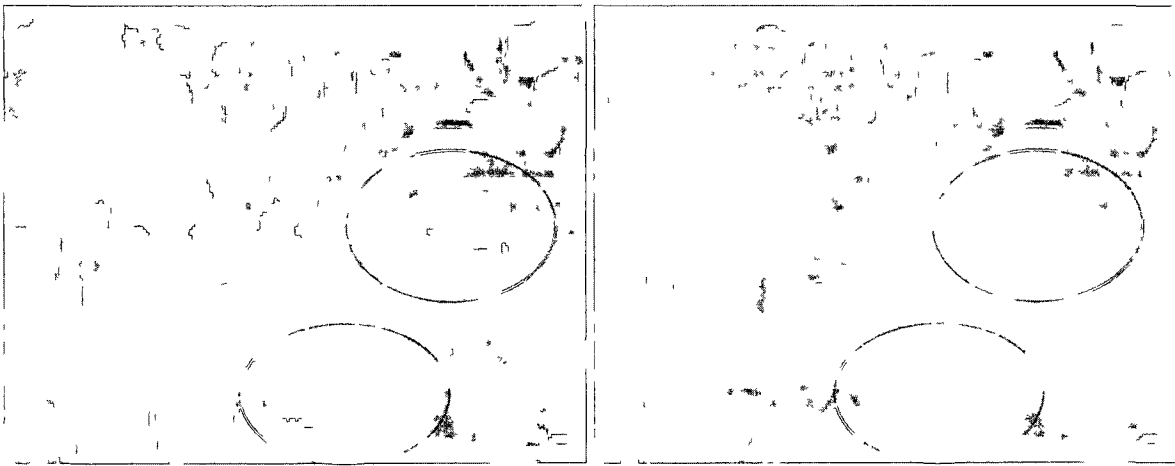
$$FliqMap(\mathbf{x}) = MVEMap(\mathbf{x}) \text{ ou } IMVMap(\mathbf{x}) \text{ ou } IPMap(\mathbf{x}) \quad (3 \ 31)$$

Soit $I_{M\alpha}(x)$, la moyenne locale des intensités de 8 voisins du pixel x . La pondération $pm(x)$ illustrée à la figure 3 13 est la version lissée de $FliqMap(x)$ définie comme suit

$$pm(x) = lp(x) * FliqMap(x) \quad (3 32)$$

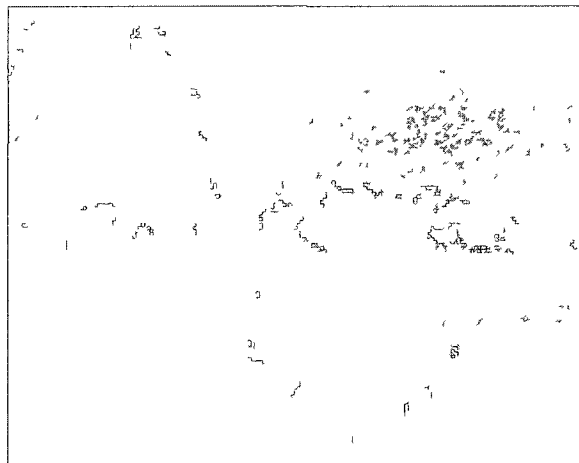
Finalement, le post-traitement multiplexe analogiquement l'image moyenne $I_{M\alpha}(x)$ et celle de sortie $Io_{1-n-\alpha}(x)$ de la section précédente 3 3 1 selon le signal filtré $pm(x)$

$$Io_{n-\alpha}(x) = (I_{M\alpha}(x) - Io_{1-n-\alpha}(x)) pm(x) + Io_{1-n-\alpha}(x) \quad (3 33)$$



a) Image de sortie de section 3 3 1

b) Image de sortie du post-traitement



c) Image de la différence multipliée par 3

Figure 3 14 Post-traitement pour scintillement de la séquence *Cheerleaders*

La figure 3 14 illustre un résultat du travail de post-traitement pour la trame 59^e de la séquence *Cheerleaders*. Il semble que l'image 3 14b est un peu plus floue que celle de 3 14a. Cependant, l'image 3 14c montre une différence entre celle avant et celle après le traitement, qui se situe seulement aux contours des pompons. L'image 3 14b est plus naturelle que celle de 3 14a aux bordures des pompons qui sont anormalement trop nettes.

Le défaut de l'interpolation d'image autour de la transition de deux zones de différents vecteurs de mouvement est très visible en regardant une seule image. Pratiquement, ce défaut est souvent apparu dans la zone d'occlusion d'un objet passant assez rapide. Il est bon de noter que, normalement, les résultats ne sont pas très différents entre la sortie $I_{o1_{n-\alpha}}(\mathbf{x})$ de la section 3 3 1 et celle $I_{o_{n-\alpha}}(\mathbf{x})$ de la section 3 3 2 en les regardant en temps réel. Cependant, en examinant « image par image », le post-traitement est vraiment efficace pour les scènes difficiles où le mouvement d'objets est rapide et arbitraire, ce qui rend imprécise l'estimation de mouvement et, par la suite, l'interpolation.

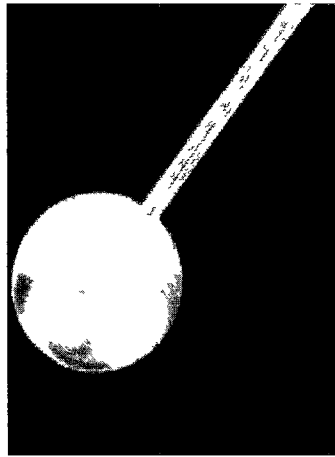
3.4 Résultat

La présente section est placée ici pour compléter les résultats partiellement présentés dans les deux sections précédentes. Nous aborderons la réduction de saccade, la réduction de l'effet de bloc dans l'interpolation finale et la réduction de halo. Comme on a mentionné au chapitre 2, plusieurs techniques d'interpolation existant utilisent les filtres temporels pour rendre flou le résultat lorsque les vecteurs de mouvement sont estimés comme non fiables. Dans cette section, les titres « réduction de saccade » seront associés avec « réduction de flou ».

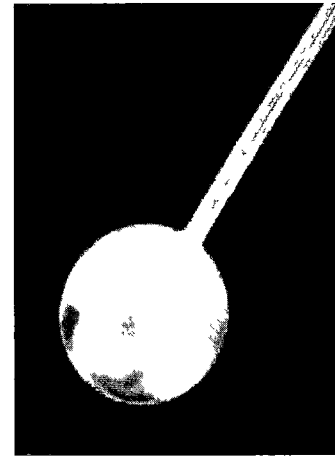
3.4.1 Réduction de saccade et de flou

Pour certains extraits (*clips*) de vieux films de faibles taux d'images, le phénomène de saccade est très prononcé lors de la présence de mouvements assez réguliers. Les « *pan*, *scan* et *zoom* » des caméras sont des exemples typiques. La figure 3.15 suivante présente le mouvement d'une pendule et ses images interpolées par la technique proposée.

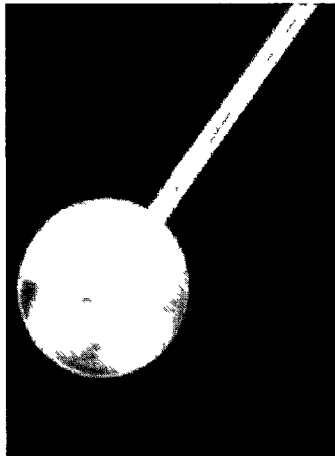
Parmi les séquences dans le tableau 3.2 à la fin de la section, deux séquences illustrent plus clairement la réduction de saccade et de flou : *Cheerleaders Pan* et *Flower Field*.



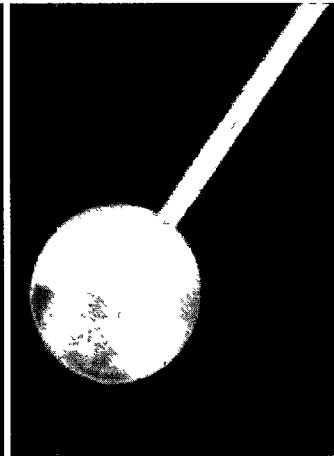
a I_{n-1}



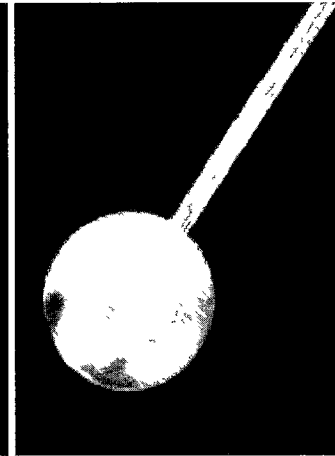
b I_n



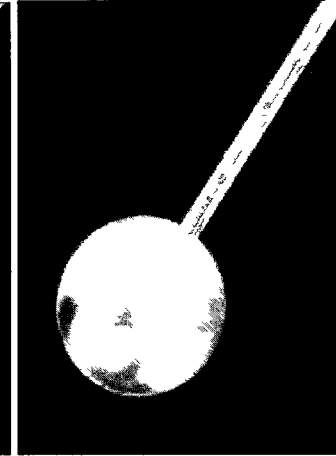
c $I_{n-4/5}$



d $I_{n-3/5}$



e $I_{n-2/5}$



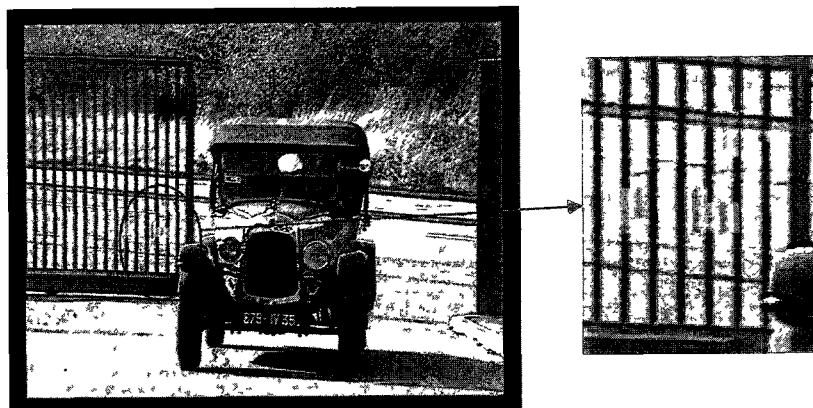
f $I_{n-1/5}$

Figure 3 15 Réduction de saccade de la séquence *Pendule*

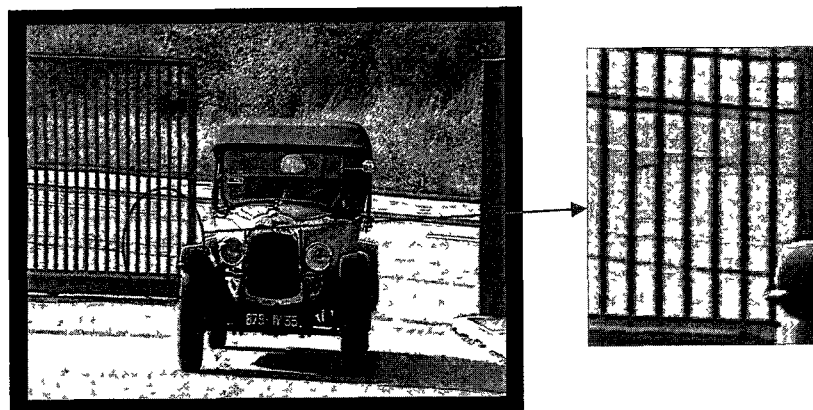
3.4.2 Réduction de l'effet de bloc

La figure 3 16 suivante, *Car Gate*, illustre la réduction de l'effet de bloc dans l'arrière-plan. Dans cette séquence, le mouvement de l'objet (voiture) est faible. L'arrière-plan, partie de la chaussée et des herbes, est relativement lisse, l'effet de bloc n'est pas perceptif. Par contre, la partie de la porte est relativement périodique, l'estimation par bloc peut résulter des vecteurs de mouvement erronés comme l'illustre la figure 3 17a

L'algorithme proposé MVS pour la sélection des vecteurs de mouvement permet la correction de ces défauts (figure 3 17b). D'une façon similaire, la figure 3 9 de la séquence *Bus* à la section 3 2 illustre le même effet.



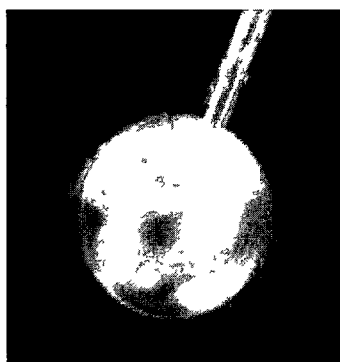
a Méthode conventionnelle



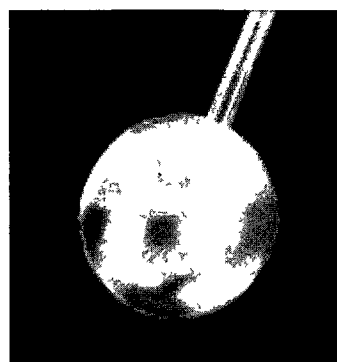
b Méthode proposée

Figure 3 16 Réduction de l'effet de bloc sur l'arrière-plan de la séquence *Car Gate*

La figure 3 17 suivante illustre l'effet de bloc qui peut se manifester différemment autour d'un objet en mouvement



a Méthode conventionnelle



b Méthode proposée

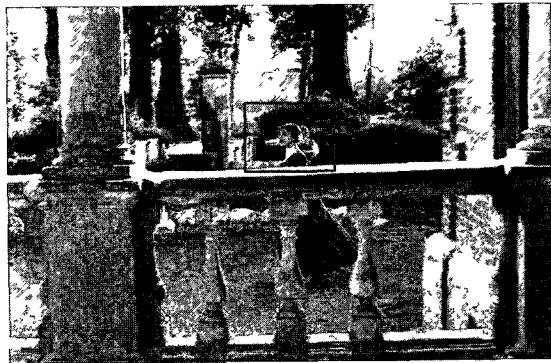
Figure 3 17 Réduction de l'effet de bloc autour de la pendule en mouvement, FRC 5x, $\alpha = 0.4$

Les séquences dynamiques *Walking Girl* et *Mobile Calendar* au tableau 3 2 qui se trouvent dans le DVD ci-joint à l'annexe C démontreront plus clairement la réduction de l'effet de bloc

3.4.3 Réduction de l'effet de halo

Dans les méthodes conventionnelles, l'effet de halo se produit dans les zones d'occlusion lors d'un objet de l'avant-plan en mouvement. Ses vecteurs de mouvement appliqués à l'arrière-plan peuvent déformer la structure de ce dernier. La figure 3 18 suivante de la séquence *Walking Girl* illustre les résultats d'une interpolation par 5 entre une méthode conventionnelle et celle proposée. Parmi les extraits (*chips*) vidéo utilisés dans cette thèse, cette séquence démontre bien l'efficacité de la correction des vecteurs de mouvement pour la réduction de halo.

D'une façon similaire, la figure 3 19 permet une comparaison de la méthode proposée avec celle de YUVsoft [Vatolin et Grishin, 2006]



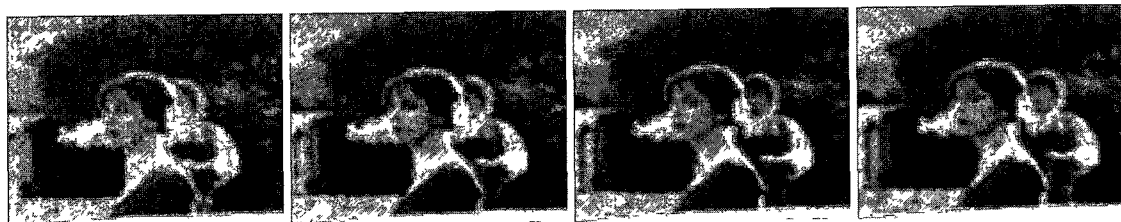
a Trame précédente



b Trame courante



c Méthode conventionnelle, FRC 5x



d Méthode proposée, FRC 5x

Figure 3 18 Réduction du halo avec la séquence *Walking Girl*

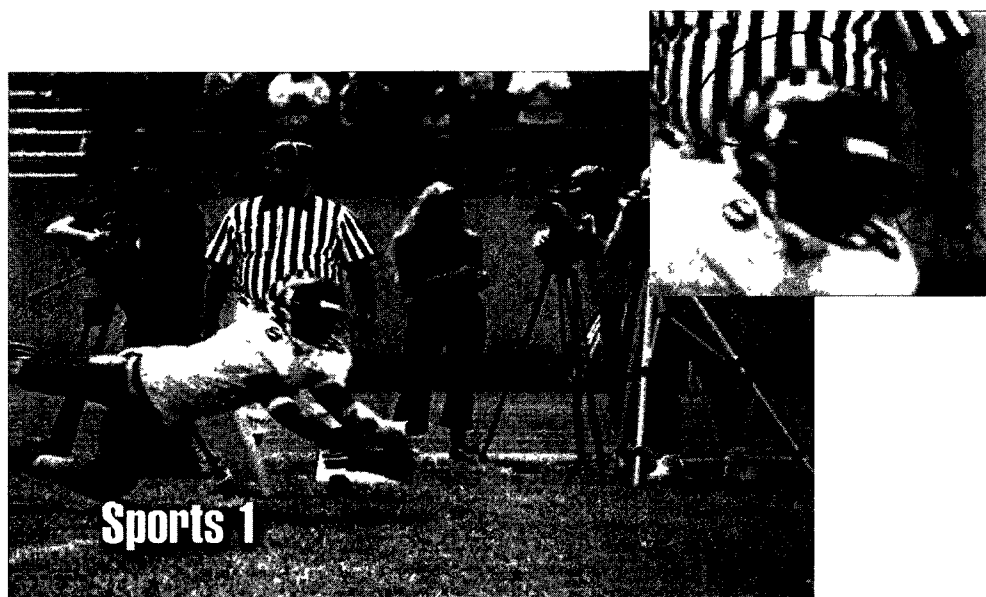


Figure 3 19 Réduction de l'effet de halo en comparaisant avec la méthode de YUVsoft [Vatolin et Grishin, 2006]

Le tableau 3 2 suivant indique les séquences vidéo interpolées dans le DVD joint à la thèse à l'annexe C afin de mieux visualiser les résultats

Tableau 3 2 Séquences incluses dans le DVD joint avec la thèse

| Nom de la séquence | Caractéristiques | Facteur d'interpolation | Réduction de l'effet de bloc | Réduction du halo | Réduction de saccade et de flou |
|------------------------------|-----------------------------------------------------------|-------------------------|------------------------------|-------------------|---------------------------------|
| <i>Cheerleaders Pan</i> | Mouvement horizontal | 4x | | | x |
| <i>Flower Field</i> | Mouvement de la camera <i>pan</i> vertical et <i>zoom</i> | 5x | | | x |
| <i>Mobile Calendar (CIF)</i> | Zone d'occlusion, mouvement complexe | 4x | x | x | |
| <i>Walking Girl</i> | Pan, et zones d'occlusion | 2x | x | x | |
| <i>Cheerleaders</i> | Zones d'occlusion, mouvement complexe | 2x | x | x | x |
| <i>Ferris Wheels</i> | Zone d'occlusion, mouvement de rotation | 4x | x | x | x |

Ces résultats complètent les travaux concernant l'interpolation temporelle pour SDTV Le prochain chapitre sera réservé à la haute définition avec l'approche de la décomposition en hiérarchie

4 APPROCHE HIÉRARCHIQUE POUR HDTV

4.1 État de l'art

Lorsque les images sont à haute ou encore à ultra définition, il ne sera plus efficace d'utiliser la recherche exhaustive pour l'estimation de mouvement. En termes de résolution en pixel, un vecteur de mouvement dans la SDTV sera agrandi approximativement par les facteurs de proportionnalité entre les formats d'images. On recourt normalement aux algorithmes sous-optimaux hiérarchiques ou similaires dans le seul but de réduire la complexité de l'estimation de mouvement.

Une estimation de mouvement hiérarchique n'est pas seulement utilisée pour la résolution à haute définition. Dépendamment du besoin de simplifier la complexité de calcul, l'estimation de mouvement hiérarchique peut aussi être utile pour les autres formats d'image comme CIF, SIF, SDTV.

La technique hiérarchique de Thoma et Bierling [Thoma et Bierling, 1989], précitée dans la section 2.2, se base sur la technique de recherche à « pas logarithmique du déplacement D » (*logD-step search technique*) avec le filtrage passe-bas.

Les techniques similaires de l'estimation de mouvement utilisant une décomposition pyramidale à trois niveaux hiérarchiques sont présentées dans [Uchida *et al*, 1997, (USP5610658)] et dans [Jung, 1998, (USP5754237)].

Pour simplifier la complexité du calcul, Chang et Yanagihara [Chang et Yanagihara, 2000, (USP6130912)] suggèrent une recherche en trois étapes avec des projections intégrales sur les axes verticaux et horizontaux de chaque super micro-bloc. Chen *et al* [Chen *et al*, 2000, (USP6160850)] emploient également des techniques de recherche en trois étapes et une unité de contrôle appropriées pour réduire la mémoire nécessaire. Dans l'application pour HDTV, Beon *et al*, dans [Beon *et al*, 2008, (USP20080074350A1)] ont proposé l'utilisation de deux processeurs pour partager le travail en découpant pratiquement l'image à interpoler en deux.

En appliquant l'idée de la méthode par appariement de bloc hiérarchique (*Hierarchical Block Matching Algorithm*, HBMA) pour l'estimation de mouvement basé bloc, Bierling [Bierling, 1988], [Wang *et al*, 2002] ouvre une classe d'algorithmes rapides de BMA. La méthode hiérarchique ou multi-résolution se base sur l'hypothèse que l'estimation de mouvement peut être calculée avec les images sous-échantillonnées. Une structure pyramidale de deux images successives d'entrée est produite par des filtrages passe-bas successifs. Bierling applique un filtre gaussien passe-bas à la référence [Bierling, 1988].

Jeon *et al* [Lee *et al*, 2003] utilisent les pyramides de Gauss et de Laplace [Burt et Adelson, 1983] pour créer la structure hiérarchique d'images. Au niveau le plus faible, l'image interpolée est créée en utilisant le vecteur de mouvement de l'estimation de mouvement forward. Ensuite, elle est sur échantillonnée afin d'avoir la même taille que l'image au niveau plus élevé. Cette image interpolée est employée afin d'obtenir l'estimation de mouvement à ce niveau. Ce processus est répété jusqu'au niveau le plus élevé. L'idée d'utilisation d'une image interpolée comme image de référence dans l'estimation de mouvement au bas niveau pour trouver les vecteurs de mouvement au niveau plus élevé est efficace pour réduire l'artefact de bloc, mais cela n'est pas suffisamment précis pour trouver le vecteur de mouvement correspondant à la bordure de l'objet.

4.2 Recherche proposée

À l'exception du premier article cité [Thoma et Bierling, 1989] pour le cas spécifique de la téléconférence, les références étudiées ne portent pas attention sur l'effet du halo dans l'estimation hiérarchique de mouvement. Pourtant, dans une décomposition d'image à plusieurs niveaux, la recherche non exhaustive avec l'imprécision impliquée d'estimation de mouvement risque d'étendre les zones de halo. De plus, lorsque l'image s'agrandit par rapport à la résolution standard, le halo l'est éventuellement aussi.

Dans le présent chapitre, après une description de la structure générale, la recherche proposée comporte les volets suivants

- D'abord, pour généraliser la technique de sélection du vecteur de mouvement basé-pixel à haute résolution, il est intéressant d'établir un nouveau critère qui permette à la

fois un raffinement du vecteur de mouvement de basse à haute résolution et un mécanisme échappatoire ou de ré-correction des vecteurs de mouvement éventuellement erronés surtout dans les zones d'occlusion

- Quant à l'interpolation d'image, il est aussi intéressant d'introduire une technique générique avec les paramètres utiles à contrôler

Les sections suivantes se basent partiellement sur le brevet en application [Tran *et al*, 2010], annexe B, dont le premier auteur est l'auteur de la présente thèse. Les travaux des autres coauteurs seront présentés sommairement dans ce chapitre s'ils sont nécessaires pour la compréhension de la conception globale du système.

Le chapitre sera organisé de la façon suivante : la section 4.3 sera une description globale du système hiérarchique proposé, la section 4.4 sera dédiée à la sélection du vecteur de mouvement basé-pixel à haute résolution, la section 4.5 sera pour une interpolation générique et finalement, les résultats de ces recherches se retrouveront dans la section 4.6.

4.3 Structure générale du système hiérarchique proposé

Le processus hiérarchique est essentiellement un arbre de décomposition d'une image originale en plusieurs sous-images de faible résolution. Après un premier filtrage et l'échantillonnage, l'image de résolution originale ou l'image de premier niveau donne une sous-image de résolution plus basse en déterminant celle de deuxième niveau. Le processus du filtrage et de la décomposition continue à s'effectuer afin d'obtenir la sous-image de 3^e niveau, et ainsi de suite, dans la décomposition pyramidale hiérarchique. Pour l'interpolation d'image, il est courant de décomposer les deux images adjacentes dans une séquence de vidéo en trois ou parfois quatre niveaux hiérarchiques.

La figure 4.1 illustre le système proposé qui emploie une approche hiérarchique de deux niveaux pour HDTV.

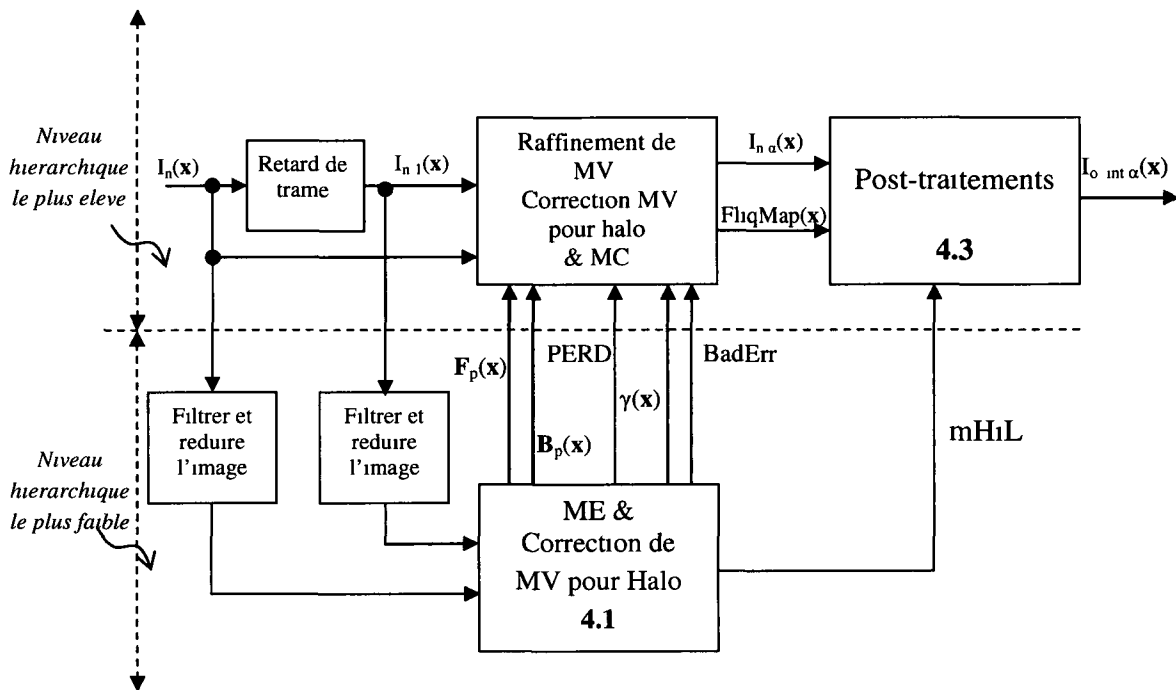


Figure 4 1 Schéma global du système HDTV

Au niveau hiérarchique le plus faible, les images à l'entrée I_n et I_{n-1} sont filtrées et sous-échantillonnées par un facteur ($U \times U$) de deux par deux (2×2) pour une image HDTV 720 ou par un facteur de trois par trois (3×3) pour une image HDTV 1080. Ces facteurs sont choisis pour garder la même dimension (640×360) pour l'image réduite à basse résolution (Pratiquement, toute image de 720 p peut être interpolée à 1080 p avant le sous-échantillonnage 3×3 pour avoir le même traitement). Les processus de recherche des vecteurs de mouvement « corrects » entre les images de basse résolution s'effectuent à l'aide du bloc 4.1 qui fait l'estimation de mouvement standard, la correction de vecteur de mouvement pour le halo (HMVR) et les traitements supplémentaires décrits au chapitre 3. Le bloc 4.1 procure aussi les signaux γ , $PERD$ et $BadErr$ définis au chapitre 3 et utiles pour le niveau hiérarchique supérieur [Tran et LeDinh, 2011].

Au niveau hiérarchique le plus élevé ou au niveau de l'image de résolution originale, les vecteurs de mouvement obtenus du niveau hiérarchique plus bas vont continuer à se raffiner dans le bloc 4.2 qui effectue ensuite, la réduction de halo, l'interpolation d'image à

compensation de mouvement et procure les signaux F_{liqMap} et $I_{n-\alpha}$ au bloc 4.3 dédié pour les post-traitements supplémentaires

Le schéma bloc du système HDTV peut être illustré avec plus de détails par la figure 4.2 plus loin. Au niveau hiérarchique le plus faible, le bloc 4.1 est divisé en quatre sous-blocs 4.1.1, 4.1.2, 4.1.3 et 4.1.4. Le premier, l'estimation du mouvement ME forward et celle ME backward, est celui présenté dans le brevet [Rossignol *et al.*, 2009, (USP0161763A1)]. Le but principal du bloc 4.1.1 est de trouver les bons vecteurs de mouvement en utilisant essentiellement la méthode de recherche exhaustive. L'estimation des vecteurs de mouvement en cas de périodicité est aussi un sujet très important dans la technique de conversion du taux d'image FRC. Cependant, cette recherche spécifique est le fruit d'un groupe de recherche, elle ne sera donc pas analysée en détail dans cette thèse. Le bloc 4.1.2 et la partie « correction des vecteurs de mouvement pour halo » du bloc 4.1.3 ont été présentés dans le chapitre 3. La « carte de détection de halo dans l'arrière-plan périodique » mHiL est produite selon l'algorithme illustré à la figure 3.11 du chapitre 3. Le signal mHiL sera ensuite sur-échantillonné pour avoir la même résolution que l'image d'entrée. Le bloc 4.1.4 procure les vecteurs de mouvement forward et backward F_p et B_p à la résolution de l'image originale en interpolant d'une façon adaptative [Tran *et al.*, 2010] les vecteurs de mouvement F_o et B_o venant de la basse résolution. L'interpolation de $(U \times U)$ qui est égale à (2×2) ou (3×3) selon la résolution originale de l'image HDTV se fait à l'aide des filtres polyphasés séparables avec un gain U de deux ou trois appropriés. L'adaptation de l'interpolation est basée à son tour sur la ressemblance locale des vecteurs de mouvement utilisés dans l'interpolation. On évite ainsi l'introduction des faux vecteurs de mouvement dans le processus de filtrage.

Quant au niveau hiérarchique le plus élevé, il compose trois sous-blocs de 4.2 (4.2.1, 4.2.2, 4.2.3) et le bloc 4.3 pour le post-traitement. Le bloc 4.2.1 dont les sorties sont $F_r(x)$ et $B_r(x)$ reçoit les entrées du niveau plus faible $F_p(x)$, $B_p(x)$ et PERD, pour raffiner les vecteurs de mouvement. Ce traitement permet à ces vecteurs de mouvement de devenir plus précis à la bordure de l'objet. Les vecteurs de mouvement résultants $F_r(x)$ et $B_r(x)$ sont appliqués au bloc 4.2.2 qui réduit à son tour l'effet de halo et ensuite élimine les vecteurs de mouvement isolés. Leurs résultats $F(x)$ et $B(x)$, les vecteurs de mouvement finaux du processus, sont appliqués au

4.2.3 pour l'interpolation de trame avec la compagne de $\gamma(\mathbf{x})$ ainsi que les deux trames natives $I_n(\mathbf{x})$ et $I_{n-1}(\mathbf{x})$

En résumé, au niveau hiérarchique le plus faible, la plupart des travaux ont déjà été décrits au chapitre 3 à l'exception des interpolations des signaux de basse à haute résolution requis pour le niveau hiérarchique supérieur. Dans le dernier étage à la résolution élevée, si la correction de vecteur de mouvement (bloc **4.2.2**) est similaire à celle de basse résolution (bloc **4.1.3**), de nouveaux algorithmes sont à développer pour le raffinement basé pixel (bloc **4.2.1**), l'interpolation générique à compensation de mouvement (bloc **4.2.3**) et finalement les post-traitements supplémentaires (bloc **4.2.4**).

La figure 4.3 illustre le système hiérarchique proposé à trois niveaux pour ultra définition. Afin de simplifier le schéma, seulement les connexions de signaux principaux y sont illustrées. À part le niveau intermédiaire, la structure à trois niveaux est similaire à celle précédente de deux niveaux hiérarchiques avec les mêmes opérations décrites. Il n'est donc pas nécessaire de donner une description détaillée pour cette configuration.

Remarque sur la complexité et la réalisabilité du système

Dans un développement classique des algorithmes, la complexité est évaluée en termes d'opérations ou encore par le temps d'exécution du processeur utilisé pour fin de simulation ou même d'exécution. Le présent projet vise par contre à fournir à un circuit intégré ASIC, conçu par un groupe de matériel, les algorithmes pouvant fonctionner en temps réel et à grande vitesse. Les contraintes de complexité sont donc différentes. Pour mieux comprendre la situation, il est d'abord désirable de décrire brièvement la structure d'un système utilisant un ASIC.

D'une façon très simplifiée, la structure comprend une mémoire externe et un ASIC qui se compose, à son tour, des unités de calcul et d'une mémoire interne (*cache memory*).

La mémoire externe, parfois indésirable en occupant une certaine superficie non négligeable dans un système, sert principalement au stockage des images d'entrée et de sortie. Un convertisseur de taux d'image requérant plusieurs images d'entrée est alors non économique. L'algorithme requis se limite alors au maximum de deux images.

La mémoire interne est destinée au stockage des résultats intermédiaires (qui peuvent être une partie des images), des lignes et des pixels de retard etc. Comme une ligne contient plusieurs pixels, les dimensions des filtres ou des fenêtres utilisés doivent avoir moins de lignes que possible.

De plus, pour un concepteur d'ASIC, les problèmes centraux sont a)- la quantité de mémoire interne et surtout b)- la largeur de bande de communication de ces trois unités citées, à savoir unité de calcul, mémoire interne et mémoire externe. La communication entre les mémoires et les unités de calcul est primordiale, car il faut avoir accès (aléatoires) aux bonnes données et recharger les résultats à bons endroits.

Dans ce sens, le contrôle de l'ASIC dicte d'une certaine façon la réalisabilité du système. À titre d'exemple, dans l'estimation de mouvement à la résolution de la télévision standard, la zone de recherche exhaustive possible est fixée à $(\pm 20 \text{ lignes}) \times (\pm 24 \text{ colonnes})$ selon un estimé de la largeur de bande de communication. Si le vecteur de mouvement dépasse cette limite, l'algorithme développé doit envisager certaines techniques de repli qui ne sont pas le cadre du présent travail. Si la même résolution de recherche exhaustive était appliquée pour la haute résolution, la dimension de la zone de recherche deviendrait neuf fois plus grande $(\pm 60 \text{ lignes}) \times (\pm 72 \text{ colonnes})$ qui devient simplement irréalisable. L'algorithme à développer doit alors recourir à une autre approche. Par exemple, celle hiérarchique proposée dans ce chapitre. Avec un sous-échantillonnage de 3×3 , pour le signal de haute résolution de 1080 lignes progressives, l'approche hiérarchique permet la réalisabilité du système tout en faisant une économie des calculs de l'estimation de mouvement par un facteur de neuf. Pour l'ultra-définition de 4K pixels de large, voir Figure 4.2, le facteur serait $(3 \times 3) \times (2 \times 2) = 36$. Finalement, comme l'approche hiérarchique est sous-optimale, il est important d'augmenter la performance du système. Autrement dit, l'algorithme à proposer doit procurer plus de résolution dans l'estimation de mouvement à partir des signaux à basse résolution dans l'approche hiérarchique.

En bref, la complexité du système proposé peut être classée comme «moyenne-grande» selon le tableau 2.1 ♦

La présente section a été réservée à la description globale de la structure hiérarchique proposée et son fonctionnement. À la prochaine section, les descriptions seront consacrées au raffinement hiérarchique basé-pixel pour vecteur de mouvement. C'est l'opération qui permet le passage des vecteurs de mouvement estimés de basse résolution à haute résolution avec une précision raffinée.

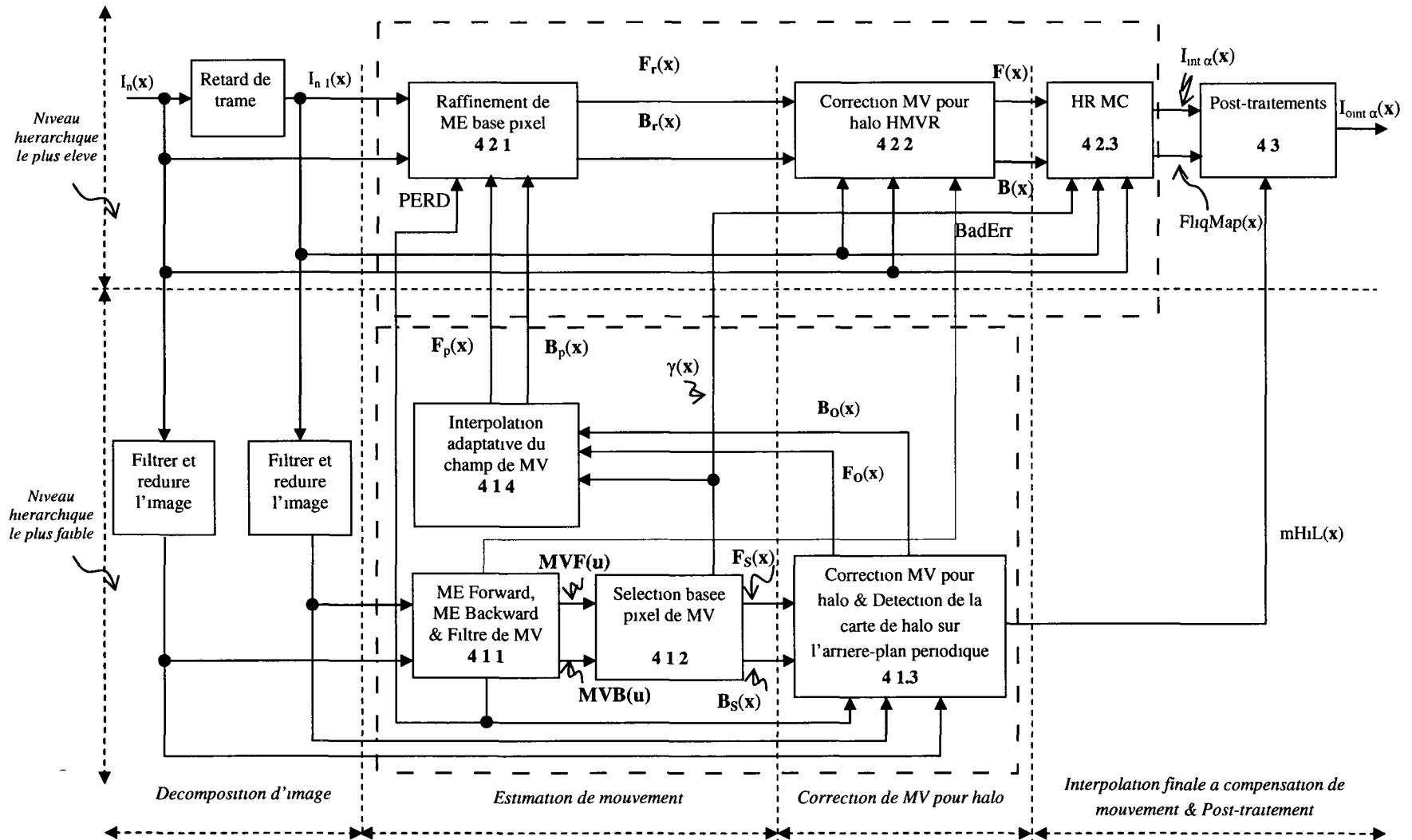


Figure 4 2 Schéma détaillé du système HDTV

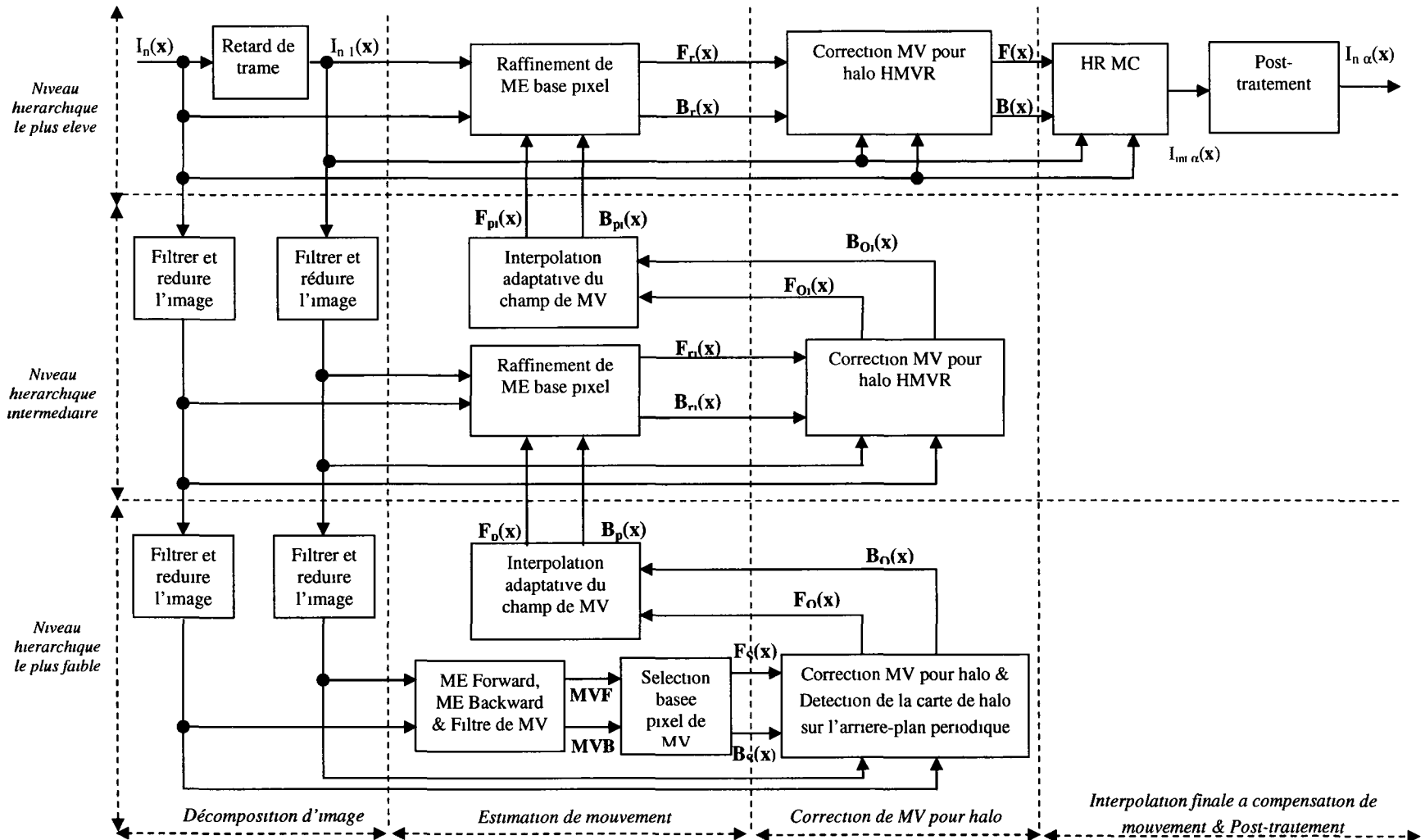


Figure 4 3 Système proposé à trois niveaux hiérarchiques

4.4 Raffinement hiérarchique basé pixel pour vecteur de mouvement

Si les vecteurs de mouvement forward et backward $F_p(x)$ et $B_p(x)$ venant de l'interpolation ($U \times U$) dans le bloc 4.1.4 sont déjà basés-pixel et de résolution appropriée pour le niveau hiérarchique supérieur, leurs précisions sont encore à réajuster d'une part. D'une part, aux alentours des zones d'occlusion, il existe probablement encore quelques pixels isolés avec un faux vecteur de mouvement. Il est alors nécessaire d'avoir un mécanisme de re-correction des vecteurs de mouvement erronés. La présente section est réservée pour ce volet de recherche.

Contrairement aux estimations de mouvement hiérarchiques classiques, comme, par exemple, la recherche à pas logarithmique, qui travaillent plutôt sur les intensités des pixels, notre approche similaire à la sélection de vecteur de mouvement présentée au chapitre 3, se concentre sur les vecteurs de mouvement.

Comme l'interpolation finale requiert les vecteurs de mouvement forward et backward. Le raffinement hiérarchique s'effectue pour chaque vecteur de mouvement. Pour fixer les idées, il sera commode de présenter l'algorithme proposé pour le cas du vecteur de mouvement forward. Les relations similaires pour le cas backward deviendront immédiates.

Pour le vecteur de mouvement forward, on considère deux ensembles de vecteurs au pixel considéré. Le premier est pour la re-correction d'un vecteur de mouvement erroné et le deuxième pour la précision.

Le premier ensemble S_{1F} (F pour forward) se compose de neuf (9) vecteurs de mouvement dont huit vecteurs de mouvement de pixels avoisinant du pixel central considéré et le vecteur de mouvement du pixel central lui-même. Toutefois, les positions des pixels avoisinants pour l'estimation de mouvement ne sont pas nécessairement les pixels adjacents immédiats, mais ceux à une distance multiple de U . $x + (kpU, kqU)$. U est choisi à 3 pour le cas HDTV 1080 et à deux (2) pour HDTV 720. Les premiers ensembles pour les vecteurs de mouvement forward et backward sont donc

$$S_{1F} = \{F(x + (kpU, kqU))\} \quad (4.1)$$

$$S_{1B} = \{B(x + (kpU, kqU))\} \quad (4.2)$$

où p et q prennent valeur dans $\{-1, 0, +1\}$ et $k = 2$. Le nombre 9 des candidats est dû à la complexité, tandis que la valeur 2 de k résulte déjà d'un test plus ou moins exhaustif. Le résultat est clair : il faut choisir les candidats proches du pixel considéré mais à l'extérieur de la surface d'expansion ($U \times U$) de l'interpolation.

La figure 4.4 suivante illustre les neuf positions marquées par un « x » de pixels sélectionnés. Leurs vecteurs de mouvement sont utilisés comme candidats possibles pour le raffinement du vecteur de mouvement du pixel considéré.

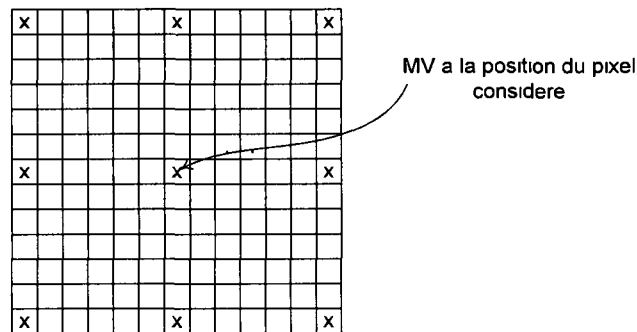


Figure 4.4 Premier ensemble des neuf vecteurs de mouvement candidats

Le deuxième ensemble S_{2F} des vecteurs de mouvement forward proposés pour fournir une précision plus élevée dans l'estimation de mouvement comprend les huit vecteurs de mouvement qui sont proches du vecteur de mouvement en cours. Pour obtenir ces vecteurs de mouvement qui sont fictifs, on ajoute les précisions supplémentaires telles que

$$S_{2F} = \{F(x) + \lambda \Delta, \Delta \neq (0, 0)\} \quad (4.3)$$

$$S_{2B} = \{B(x) + \lambda \Delta, \Delta \neq (0, 0)\} \quad (4.4)$$

Dans ces expressions, le vecteur additionnel $\Delta = (\Delta_h, \Delta_v)$ de composantes respectivement horizontale et verticale Δ_h et Δ_v qui prennent les valeurs dans $\{-1, 0, +1\}$, sauf le cas $\Delta_h = \Delta_v = 0$ simultanément. Le facteur de proportionnalité λ est choisi égal à 1 au moins pour le cas HDTV 1080.

Pour la simplicité, on définit $\mathbf{F}_\Delta = \mathbf{F}_\Delta(\mathbf{x})$ un vecteur de mouvement forward parmi les 17 vecteurs de mouvement dans les deux (2) ensembles S_{1F} ou S_{2F} décrits respectivement par (4 1) ou (4 3) On a donc

$$\mathbf{F}_\Delta = \mathbf{F}_\Delta(\mathbf{x}) \in S_{1F} \cup S_{2F} \quad (4 5)$$

$$\mathbf{B}_\Delta = \mathbf{B}_\Delta(\mathbf{x}) \in S_{1B} \cup S_{2B} \quad (4 6)$$

Le critère de sélection de vecteur de mouvement raffiné $\mathbf{F}_r(\mathbf{x})$ parmi 17 candidats est celui qui minimise les 17 sommes des différences absolues SAD dans la fenêtre 5x5 autour du pixel à haute définition considéré De la même façon pour le vecteur de mouvement backward $\mathbf{B}_r(\mathbf{x})$ Les SADs sont définies par les équations suivantes

$$SADF_\Delta(\mathbf{x}) = c_F \left(\frac{1}{25} \sum_{k_h=-2}^2 \sum_{k_v=-2}^2 |I_n(\mathbf{x} + \mathbf{k} + \alpha \mathbf{F}_\Delta) - I_{n-1}(\mathbf{x} + \mathbf{k} - (1-\alpha) \mathbf{F}_\Delta)| \right) + p_F \quad (4 7)$$

$$SADB_\Delta(\mathbf{x}) = c_B \left(\frac{1}{25} \sum_{k_h=-2}^2 \sum_{k_v=-2}^2 |I_n(\mathbf{x} + \mathbf{k} - \alpha \mathbf{B}_\Delta) - I_{n-1}(\mathbf{x} + \mathbf{k} + (1-\alpha) \mathbf{B}_\Delta)| \right) + p_B \quad (4 8)$$

Dans le but de réduire la complexité du calcul, les images interpolées requises par (4 7) et (4 8) sont obtenues par filtre séparable à deux coefficients

Les coefficients c_F , c_B et p_F , p_B sont définis par les équations suivantes

$$c_F = 2, \quad \text{si } \mathbf{F}_\Delta(\mathbf{x}) \in S_{2F} \ \& \ PER_{hd}(\mathbf{x}) = 1, \quad (4 9a)$$

$$c_F = 1, \quad \text{si non}$$

$$p_F = 1, \quad \text{si } \mathbf{F}_\Delta(\mathbf{x}) \in S_{2F} \ \& \ PER_{hd}(\mathbf{x}) = 1, \quad (4 9b)$$

$$p_F = 0, \quad \text{si non}$$

et $c_B = 2, \quad \text{si } \mathbf{B}_\Delta(\mathbf{x}) \in S_{2B} \ \& \ PER_{hd}(\mathbf{x}) = 1, \quad (4 10a)$

$$c_B = 1, \quad \text{si non}$$

$$p_B = 1, \quad \text{si } \mathbf{B}_\Delta(\mathbf{x}) \in S_{2B} \ \& \ PER_{hd}(\mathbf{x}) = 1, \quad (4 10b)$$

$$p_B = 0, \quad \text{si non}$$

Dans ces équations, $PER_{hd}(\mathbf{x})$ est la version répétée dans un bloc de (UxU) du signal $PERD([\mathbf{x}/U])$ à basse résolution Les facteurs de pondération c et les constantes de pénalité

(*penalty offset*) p procurent un certain favoritisme pour le premier ensemble S_{IF} (S_{IB}) des vecteurs de mouvement il est préférable de corriger les erreurs ou de conserver le vecteur de mouvement au pixel donné que de faire le raffinement par Δ

Bien qu'elle serve très rarement à obtenir deux ou plusieurs valeurs minimales de $SADF_{\Delta}$, une table de priorité est donnée dans [Tran *et al*, 2010] pour résoudre l'ambiguïté

La figure 4.5 suivante détaille le bloc 4.2.1 dans la figure 4.2 et illustre le processus de raffinement des vecteurs de mouvement forward et backward basés-pixel pour le niveau hiérarchique le plus élevé

En se référant à la figure 4.2, les vecteurs de mouvement raffinés $F_r(x)$ et $B_r(x)$ sont appliqués à leur tour au bloc 4.2.2 qui cherche les vecteurs de sortie $F(x)$ et $B(x)$ permettant une réduction possible de l'effet du halo restant À part des entrées appropriées à haute résolution, le principe de la réduction de halo demeure l'algorithme HMVR présenté au chapitre 3 Il semblera superflu de le répéter ici

Les derniers vecteurs de mouvement $F(x)$ et $B(x)$ sortis du bloc 4.2.2 seront utilisés pour l'interpolation d'image

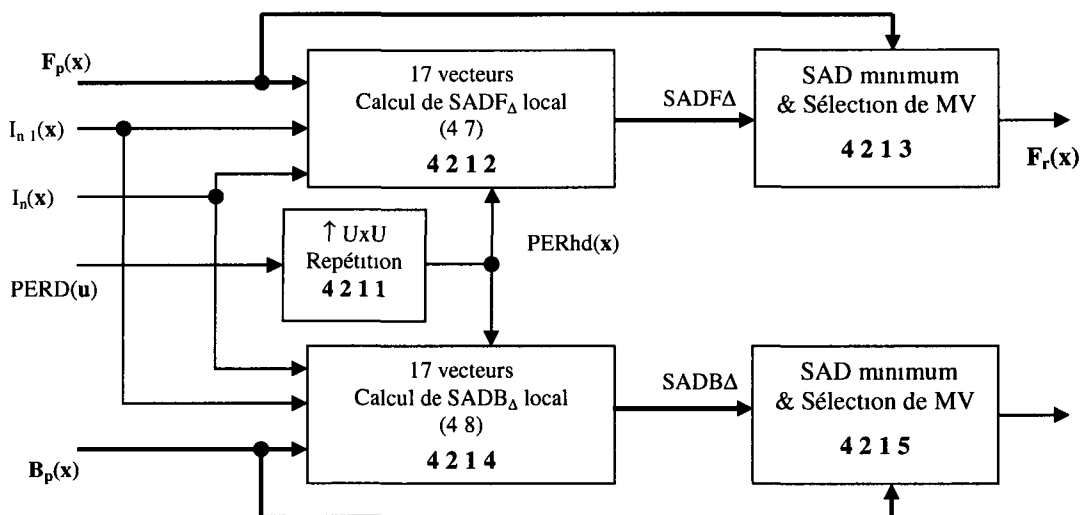


Figure 4.5 Raffinement hiérarchique basé pixel pour les vecteurs de mouvement

4.5 Interpolation d'image générique proposée

4.5.1 Révision des concepts pour la compensation de mouvement

Dans le contexte d'un FRC, c'est l'interpolation à compensation de mouvement qui donne l'image finale ou quasi finale si on tient compte des post-traitements subséquents. Il nous est donc intéressant de faire une simple révision des concepts pour la compensation de mouvement avant de suggérer une interpolation d'image.

Dans cette révision, pour limiter la longueur de la discussion, les techniques particulières de compensation de mouvement comme celles qui relient les estimations de mouvement utilisant des blocs avec recouvrement (*Overlapped block Motion Compensation, OBMC*) [Choi *et al*, 2000, Ha *et al*, 2004] ou encore le filtrage médian [De Haan, 2000] des trois interpolations ne seront pas considérés.

On s'intéresse à l'interpolation de base, qui se résume dans l'équation suivante

$$I_{n-\alpha}(x) = (1-b) I_n(x + d_F F) + b I_{n-1}(x + d_B B) \quad (4.11)$$

Dans cette équation, F et B sont respectivement les vecteurs de mouvement forward et backward au point considéré x . Le paramètre d'amplitude b sert à pondérer l'image interpolée $I_{n-1}(x + d_B B)$ et son complément $(1 - b)$ pour $I_n(x + d_F F)$. D'une façon similaire, le paramètre d_F sert à pondérer le vecteur de mouvement F et celui d_B pour le vecteur de mouvement B . Normalement, d_F et d_B sont associés à la position temporelle α de l'image à interpoler $I_{n-\alpha}(x)$. Les paramètres b , d_F et d_B varient entre 0 et 1 inclusivement.

Pour créer l'image au milieu des deux images existantes, les auteurs de [Fujiwara et Taguchi, 2005, Lee *et al*, 2007] et [Huang et Nguyen, 2008] ont suggéré la compensation de mouvement conventionnel avec simplement $I_n(x + F/2)$ ou encore la moyenne de deux interpolations forward et backward $(\frac{1}{2})I_n(x + \frac{1}{2} F) + \frac{1}{2} I_{n-1}(x + \frac{1}{2} B)$, i.e. $b = d_F = d_B = \frac{1}{2}$ comme dans [De Haan, 2000] et [LeDinh, 2003]. La première qui utilise seulement le vecteur de mouvement forward ne résout pas le problème de couvrement. La dernière qui fait la moyenne de deux images rend floue la zone d'occlusion. Bien sûr, les vecteurs de mouvement dans les publications citées sont basés bloc.

Plus général, dans [Thoma et Bierling, 1989], les auteurs ont utilisé $b = d_F = 1 - d_B = \alpha$, $\mathbf{F} = -\mathbf{B} = \mathbf{D}$ dans la zone de non -occlusion. Autrement dit, la notion de la position (α) de l'image interpolée finale $I_{n-\alpha}(\mathbf{x})$ est introduite à la fois dans la pondération de vecteur de mouvement et celle d'amplitude des images interpolées intermédiaires $I_{n-1}(\mathbf{x} + d_B \mathbf{B})$ et $I_n(\mathbf{x} + d_F \mathbf{F})$. L'introduction de la position α est méritoire pour une double raison : conserver la continuité des images et la consistance des vecteurs de mouvement, lorsque α est aux limites égales à 0 ou 1.

Dans [Kang *et al*, 2007, Choi et Ko 2007, Ling et Zhang 2008, Blanchfield *et al*, 2006], pour la pondération d'image, les auteurs ont proposé une interpolation selon la fiabilité (estimée) de chaque estimation des mouvements possibles à travers le pixel considéré. En général, la fiabilité est définie par l'inverse de la somme des différences absolues normalisée par la somme des inverses. Par exemple, en utilisant les SAD_F et SAD_B on écrit

$$b = \frac{\frac{1}{SADB}}{\frac{1}{SADF} + \frac{1}{SADB}} = \frac{SADF}{SADF + SADB} \quad (4.12a)$$

$$1-b = \frac{\frac{1}{SADF}}{\frac{1}{SADF} + \frac{1}{SADB}} = \frac{SADB}{SADF + SADB} \quad (4.12b)$$

Autrement dit, dans l'équation générique (4.11), b et son complément $(1 - b)$ valent respectivement la fiabilité du vecteur \mathbf{B} et celle de \mathbf{F} . Sans compter des divisions requises pour la normalisation, la difficulté dans ces approches réside dans le fait que SAD n'est pas nécessairement une mesure fiable, lorsque le point considéré se trouve dans les régions d'occlusion. Dans [De Haan, 2000], les vecteurs de mouvement sont aussi proposés pour l'adaptation des coefficients d_F et d_B . Encore une fois, les vecteurs de mouvement ne constituent pas nécessairement des indices fiables dans les zones d'occlusion.

Dans la même optique de fiabilité, au chapitre 3 ou dans [Tran et LeDinh, 2011], nous avons proposé $b = \gamma(\mathbf{x})$ une mesure de fiabilité, selon le contexte local, qui est définie par le rapport des pixels dans une fenêtre glissante, favorables à l'interpolation avec le vecteur \mathbf{B} par rapport au nombre total des pixels dans la fenêtre. La mesure $\gamma(\mathbf{x})$ répond bien à la qualité dans la

plupart des cas, sauf lorsque la zone d'occlusion est trop large par rapport à la dimension fixe de la fenêtre, l'artefact pourra encore être présent. De plus, les équations (3 18), (3 19) suggèrent d'utiliser une seule image existante I_n ou I_{n-1} lorsque la position α s'approche à 0 ou 1. L'équation (3 20) requiert par contre deux images I_n et I_{n-1} lorsque α se trouve autour de la position centrale $\frac{1}{2}$. Le désavantage possible de cette approche au point de vue matériel est de pouvoir supporter les deux images I_n et I_{n-1} interpolées avec plusieurs combinaisons des différents vecteurs de mouvement lors d'une interpolation multiple de plusieurs images. Ce support n'est pas commode pour la largeur de bande de communication entre la mémoire externe contenant les deux images et le processeur interne pour l'interpolation d'image.

De plus, pour juger le résultat d'un FRC, on fait appel souvent à des évaluations subjectives. Et à notre surprise, selon nos tests informels internes et selon les recommandations des clients, plusieurs observateurs ne sont pas tout à fait pour la monotonie des mouvements linéaires lisses. En d'autres termes, ces personnes préfèrent une certaine « similitude au film » (*film look*) avec les mouvements plus ou moins saccadés comme dans un film. Bien sûr, il y en a plusieurs autres qui aiment le contraire : les séquences avec les mouvements lisses.

En résumé, selon l'état de l'art, le paramètre b peut être une fonction de la position α d'une part et d'une certaine mesure de fiabilité d'autre part. Quant aux coefficients de pondérations d_F ou d_B des vecteurs de mouvement, tout semble indiquer que d_F est égal à la position α et d_B au complément $(1-\alpha)$ pour conserver l'évolution linéaire des mouvements dans le temps. Finalement, selon les tests subjectifs, certains observateurs préfèrent les mouvements lisses, d'autres non.

Devant cette situation, il est bon d'avoir une interpolation à compensation de mouvement plus générique que celles existantes, qui répond aux items suivants :

- Rendre contrôlables le paramètre d_F et son complément d_B selon la préférence subjective
- Combiner dans le paramètre b , la position α de l'image interpolée et la fiabilité
- Réduire si possible, à l'aide de l'interpolation, l'effet de halo et de scintillement tout en conservant la netteté de l'image de sortie

4.5.2 Interpolation à compensation de mouvement générique proposée

La figure 4 6 suivante illustre le cas d'une interpolation par 5. Si d_F est mis égal à α , en suivant uniformément le vecteur de mouvement, le déplacement temporel d'un objet varie linéairement en fonction de α (figure 4 6a). La figure 4 6a illustre aussi en ligne pointillée le cas d_F binaire qui correspond à la répétition des trames au lieu d'une interpolation à la compensation du mouvement.

Pour créer un mouvement non uniforme (plus saccadé) dans la séquence des images interpolées uniformément distancées, le coefficient de pondération d_F doit donc être exprimé comme une fonction générique de la distance α .

$$d_F = f(\alpha) \quad (4.13)$$

Le coefficient complémentaire d_B vaut

$$d_B = 1 - f(\alpha) \quad (4.14)$$

Le cas linéaire $f(\alpha) = \alpha$ devient alors spécifique mais demeure important dans le contexte d'un FRC.

La fonction $f(\alpha)$ doit respecter au moins les restrictions suivantes

$$f(0) = 0 \quad (4.15a)$$

$$f(1) = 1 \quad (4.15b)$$

$$0 \leq f(\alpha) \leq 1 \quad (4.15c)$$

La figure 4 6b illustre aussi l'effet de pondération non-linéaire $f(\alpha)$ sur le vecteur de mouvement forward, qui permet d'approcher l'image existante la plus proche. Par exemple, pour $\alpha = 0.2$, si on choisit $f(\alpha) = 2\alpha^2$, la pondération du vecteur F sera $f(0.2) = 0.08$ comme l'interpolation s'effectue à une nouvelle et fictive distance $\alpha' = 0.08$ qui devient encore plus proche de l'image existante I_n considérée comme sans défaut. Mathématiquement dans l'équation (4.11), $I_n(x + d_F F)$ tend vers $I_n(x)$. Il y aura donc moins d'artefact de halo.

Si on accepte le concept de rapprochement des images existantes, alors par rapport au point central ($\alpha=1/2$, $f(\alpha)=1/2$), la fonction $f(\alpha)$ est alors convexe pour α entre 0 et $1/2$ et concave pour

α entre $\frac{1}{2}$ et 1 Par conséquent, dans les images interpolées, le mouvement est doux aux alentours des images existantes mais plus saccadé au milieu

Il y a plusieurs fonctions $f(\alpha)$ possibles Dans notre simulation, on utilise la fonction quadratique $f(\alpha)$ suivante qui est continue mais définie différemment dans deux intervalles

$$f(\alpha) = 2\alpha^2 \quad \text{si } 0 \leq \alpha \leq \frac{1}{2}, \quad (4.16a)$$

$$f(\alpha) = -1 + 4\alpha - 2\alpha^2 \quad \text{si } \frac{1}{2} \leq \alpha \leq 1 \quad (4.16b)$$

C'est bien de noter que le doublage de fréquence d'images par interpolation utilise toujours le mode linéaire, $f(\frac{1}{2}) = \frac{1}{2}$

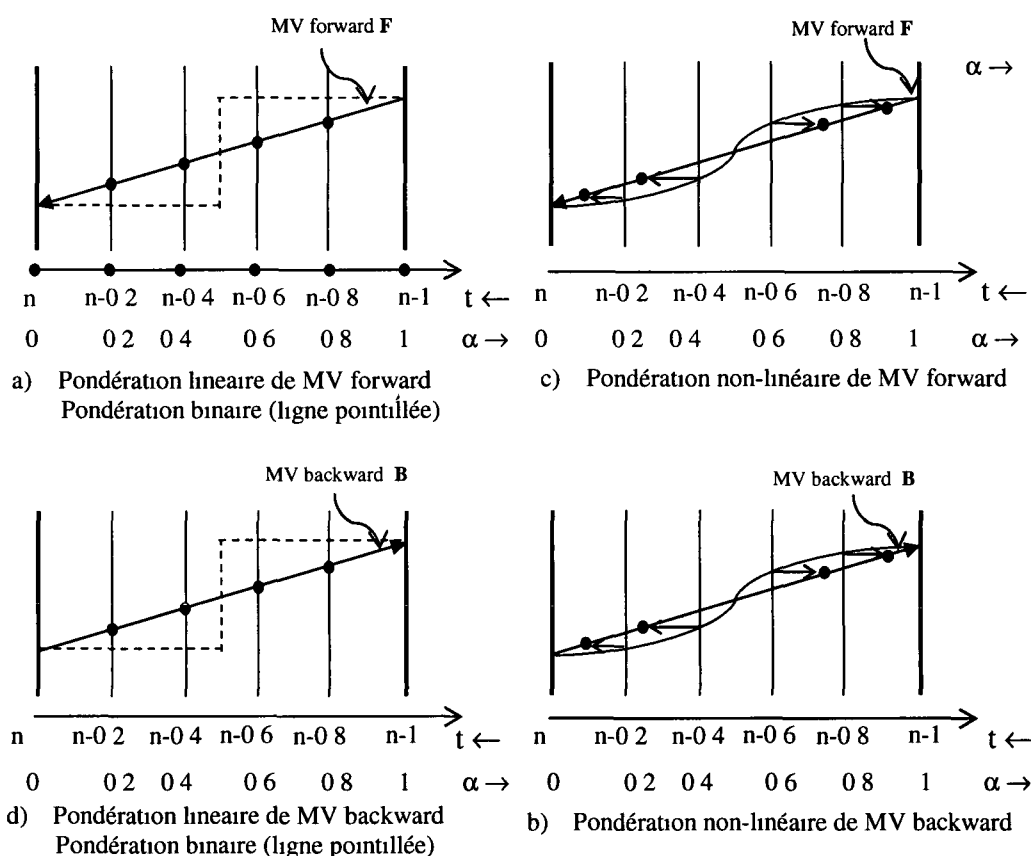


Figure 4.6 Pondération linéaire et non-linéaire des vecteurs de mouvement

La pondération non-linéaire proposée est en fait un compromis entre la pondération linéaire et la pondération binaire de la répétition des images Si la pondération linéaire est parfaite pour la

majorité des cas, le mode non-linéaire réduira significativement l'effet de halo et la déformation éventuelle d'objets dans la zone d'occlusion. Par contre, ce mode réduit moins la saccade que le mode linéaire mais crée un effet « film similaire » pouvant plaire à certains observateurs. Il n'est pas difficile de prévoir un mécanisme permettant à l'utilisateur de choisir son mode préféré. De plus, en pratique, la fonction de pondération est générée par une table (LUT) où pour chaque position α , on donne une valeur $f(\alpha)$.

On vient de présenter comment rendre contrôlables les pondérations d_F et d_B des vecteurs de mouvement dans l'interpolation (4.11). Le principe pour créer la pondération b et son complément $(1-b)$ pour les images interpolées intermédiaires $I_{n-1}(\mathbf{x} + d_B \mathbf{B})$ et $I_n(\mathbf{x} + d_F \mathbf{F})$ est très similaire. En effet, si b doit être à la fois une fonction de la distance α et de la fiabilité $\gamma(\mathbf{x})$, on peut commencer avec une fonction convexe similaire à la fonction $f(\alpha)$ dans l'intervalle $[0, 1/2]$ et la pondérer avec $\gamma(\mathbf{x})$. Notons que pour $\alpha = 0$, $b(\mathbf{x}) = 0$ et pour $\alpha = 1/2$, $b(\mathbf{x}) = \gamma(\mathbf{x})$. Ensuite, on en déduit l'expression de $b(\mathbf{x})$ pour l'intervalle $[1/2, 1]$ avec $\alpha = 1/2$, $b(\mathbf{x}) = \gamma(\mathbf{x})$ et $\alpha = 1$, $b(\mathbf{x}) = 1$. Certaines fonctions sont proposées dans [Tran *et al*, 2010]. On peut suggérer ici la paire des fonctions quadratiques suivantes qui combinent les équations (4.16a) et (4.16b) précédentes et la pondération par $\gamma(\mathbf{x})$ en suivant la procédure décrite ci-haut.

$$b(\mathbf{x}) = 4\alpha^2 \gamma(\mathbf{x}), \quad \text{si } 0 \leq \alpha \leq 1/2, \quad (4.17a)$$

$$b(\mathbf{x}) = 4(1-\alpha)^2 [\gamma(\mathbf{x})-1] + 1, \quad \text{si } 1/2 \leq \alpha \leq 1 \quad (4.17b)$$

La fonction de mixage $b(\mathbf{x})$ des images est donc complète contenant à la fois l'information de distance et de fiabilité locale.

Numériquement pour le même exemple que celui précédent avec $\alpha = 0.2$, l'image interpolée dans l'équation (4.11) devient

$$I_{n-\alpha}(\mathbf{x}) = (1-0.16\gamma) I_n(\mathbf{x} + 0.08\mathbf{F}) + (0.16\gamma) I_{n-1}(\mathbf{x} + 0.92\mathbf{B}) \rightarrow I_n(\mathbf{x} + 0.08\mathbf{F}) \quad (4.18)$$

et ce, d'une façon quasi indépendante de \mathbf{F} , \mathbf{B} et γ .

L'interpolation finale combinant à la fois les pondérations proposées d_F et b contient moins de différences temporelles. L'effet de scintillement est donc réduit.

Pour une raison économique en matériel, l'interpolation (4 11) est réécrite avec plus de détails comme suit

$$I_{n-\alpha}(x) = I_n(x + d_F F(x)) + b(x) [I_{n-1}(x + (1 - d_F) B(x)) - I_n(x + d_F F(x))] \quad (4 19)$$

La figure 4 7 illustre un schéma bloc de l'interpolateur à compensation de mouvement selon l'équation (4 19) Pour avoir plus de souplesse dans le contrôle de la pondération $b(x)$ d'images, une deuxième fonction $f(\alpha)$ linéaire ou non, mais indépendante de la première, est placée en aval de la fonction $b(x)$ [Tran *et al*, 2010] Dans ce cas, avec les mêmes $f(\alpha)$ mais indépendantes entre elles et avec $\alpha = 0.2$, l'équation (4 19) donne

$$I_{n-\alpha}(x) = (1 - 0.256\gamma) I_n(x + 0.08F) + (0.256\gamma) I_{n-1}(x + 0.92B) \rightarrow I_n(x + 0.08F) \quad (4 20)$$

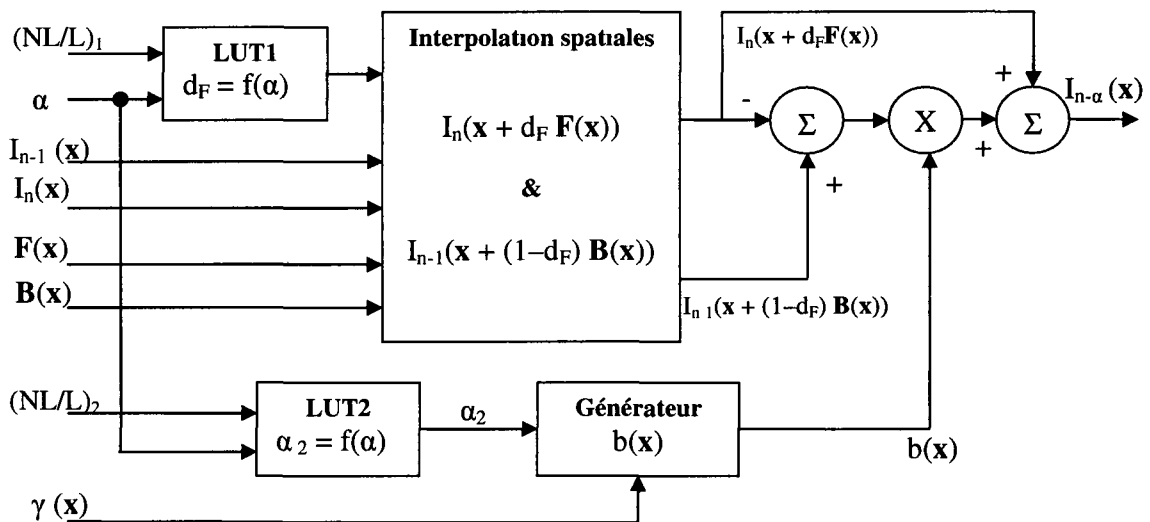


Figure 4 7 Schéma-bloc de l'interpolateur proposé

En observant l'équation (4 18), il est clair que l'équation (4 20) accélère la convergence

Le schéma comprend deux LUT indépendants pour $f(\alpha)$, un générateur pour $b(x)$, un système d'interpolations spatiales pour $I_n(x + d_F F(x))$ et $I_{n-1}(x + (1-d_F) B(x))$ et un mixeur doux pour l'image interpolée $I_{n-\alpha}(x)$

Quant aux interpolations spatiales pour $I_n(x + d_F F(x))$ et $I_{n-1}(x + (1-d_F) B(x))$, elles sont effectuées par les filtres séparables. Le facteur de sur échantillonnage utilisé est de 16x16 qui

donne une précision, par dimension, d'un seizième (1/16) de pixel. Ces filtres, en point fixe, réalisés polyphasés avec quatre coefficients par phase donnent d'excellents résultats pour SDTV et HDTV.

L'interpolation finale proposée est générique et applicable pour toute bonne estimation du mouvement et de la fiabilité.

Finalement, comme dans le cas de SDTV, certains traitements supplémentaires comme la réduction de halo dans l'arrière-plan périodique et quasi statique et la réduction de scintillement sont utilisables pour HDTV. Cependant, à l'exception de la résolution plus élevée, ces techniques utilisant les mêmes concepts que ceux présentés à la section 3.3, ne seront pas décrites à nouveau ici.

4.6 Résultats

La présente section est consacrée à démontrer l'efficacité des algorithmes dans la structure hiérarchique proposée. Il s'agit alors de la réduction de halo, de la résolution apportée pour les vecteurs de mouvement dans les niveaux hiérarchiques supérieurs, de l'interpolation générique utilisant les positions non-linéaires fictives, et de la réduction de saccade.

Comme les images sur papier sont statiques, il peut être alors difficile de démontrer les effets comme le halo ou encore la réduction de saccade. Ce sont des effets qui surgissent temporellement lors du déroulement des images sous les yeux. Comme méthodologie, nous nous proposons d'utiliser les différents vecteurs de mouvement à la sortie des blocs 4.1.1, 4.1.2, 4.2.1 et 4.2.2 illustrés à la figure 4.2 pour interpoler l'image. Les images résultantes permettent ainsi de suivre les améliorations apportées à chaque sous-bloc et/ou de mesurer les performances obtenues. Cette méthode convient pour les deux effets cités en rubrique. Par contre, la réduction de saccade surtout pour une interpolation multiple de plusieurs images requerrait une visualisation temporelle, un DVD ci-joint est donc réservé à cet effet.

Avant de présenter les résultats, il est bon de rappeler le but de chaque sous-bloc en question. Le bloc 4.1.1 sert à combiner et filtrer les vecteurs de mouvement forward et backward [Rossignol *et al.*, 2010] provenant de l'estimation de mouvement basée bloc classique, le

bloc **4.1.2** sert à réduire l'effet de bloc en sélectionnant chaque pixel parmi les vecteurs de mouvement entourant un vecteur de mouvement forward et un vecteur de mouvement backward appropriés, le bloc **4.2.1** sert à ré-estimer les vecteurs de mouvement au niveau hiérarchique supérieur, et finalement, le bloc **4.2.2** sert à réduire l'effet de halo au dernier niveau hiérarchique

De plus, pour fin de comparaison, les images sont interpolées, comme nous avons mentionné en rubrique, à la résolution des image à haute définition à l'entrée Dans ce cas, les vecteurs de mouvement de basse résolution des blocs **4.1.1** ou **4.1.2** sont répétés ou interpolés par $U \times U = 3 \times 3$ ce qui rend la dimension 8×8 d'un bloc de standard résolution de base à 24×24 en haute définition

Dans la présente section, nous utiliserons la séquence *Walking Girl* pour halo et artefact de bloc, celle de *New York City Spin* pour la résolution de vecteur de mouvement et *Big Buck Bunny* et *Home in Park* pour l'interpolation générique proposée Les séquences *Stairs* et *Scrolling Text* se trouvent dans le DVD de l'annexe C pour démontrer la réduction de saccade et de flou

4.6.1 Halo et Artefact de bloc dans la structure hiérarchique

La figure 4 8 à la page suivante pour la séquence de *Walking Girl* illustre le rôle de chaque sous-bloc proposé dans la structure hiérarchique Les figures 4 8a et 4 8b correspondent aux images originales I_n et I_{n-1} mais découpées autour de la région d'intérêt La figure 4 8c illustre bien l'effet de bloc dans l'image interpolée $I_{n-\alpha}$ et délimitée dans la même région d'intérêt La figure 4 8d illustre la réduction de bloc et aussi l'imprécision des vecteurs de mouvement autour de la transition des régions La figure 4 8e illustre la correction de ces imprécisions en réestimant les vecteurs de mouvement au niveau hiérarchique supérieur La figure 4 8f illustre la réduction de halo après le bloc **4.2.2** Pour remarquer la réduction de halo avec les images statiques, il faudrait examiner les détails de l'arrière -plan autour de l'objet en mouvement, et ce, avant et après la réduction de l'effet Dans cette séquence, ce sont les régions autour de la face et du cou de la jeune fille entre les deux figures 4 8e et 4 8f La séquence dans le DVD compare la différence entre block **4.1.1** et **4.2.3**



a Image originale découpée de I_{n-1}



b Image originale découpée de I_n



c Image interpolée $I_{n-1/2}$
avec MVs sortis du bloc 4.1.1



d Image interpolée $I_{n-1/2}$
avec MVs sortis du bloc 4.1.2



e Image interpolée $I_{n-1/2}$
avec MVs sortis du bloc 4.2.1



f Image interpolée $I_{n-1/2}$
avec MVs sortis du bloc 4.2.2

Figure 4 8 Réduction de l'effet de halo et de bloc avec la séquence *Walking Girl*

4.6.2 Résolution apportée pour les vecteurs de mouvement au niveau hiérarchique supérieur

Pour mieux illustrer le résultat, le test est effectué sur la séquence *NewYork City Spin*. Il s'agit d'une vue aérienne de la ville avec plein de détails de la structure de la ville mais tournant autour du point donné.

En regardant la séquence interpolée en temps réel, on s'aperçoit que la structure de la ville semble vibrer d'une façon hésitante lorsque les vecteurs de mouvement n'ont pas suffisamment de précision autrement dit lorsqu'ils ne sont pas raffinés de basse à haute résolution. Pour mesurer la performance apportée des algorithmes proposés, on ré-interpole les images paires à partir des images impaires et on compare les résultats avec les images paires originales.

Les résultats sont représentés dans les figures 4.9 et 4.10. La figure 4.9 illustre la trame originale avec l'indication de la zone d'intérêt découpée. La figure 4.10 illustre successivement les images ré-interpolées et leur image d'erreur respective par rapport à l'image originale. Précisément, la figure 4.10a correspond aux vecteurs de mouvement sortis du bloc 4.1.1, la figure 4.10b correspond au bloc 4.1.2, la figure 4.10c au bloc 4.2.1 mais $\lambda = 0$ et, finalement la figure 4.10d au bloc 4.2.1 mais $\lambda = 1$. Le paramètre λ est défini par les équations (4.3) et (4.4) pour le raffinement des vecteurs de mouvement.

Comme on peut constater, les images d'erreur représentent bien les améliorations apportées. De plus, la figure 4.11 illustre les erreurs de chaque étape. Quant aux mesures métriques, le tableau 4.1 suivant résume l'évolution des PSNR des images interpolées de chaque étape de la séquence *NewYork City Spin*.

Tableau 4.1 Évolution des PSNR-*NewYork City Spin*

| Numéro de l'étape | PSNR en dB | Variation de PSNR |
|---------------------------------|------------|-------------------|
| #1 Bloc 4.1.1 | 27.83 | x |
| #2 Bloc 4.1.2 | 28.92 | 1.09 |
| #3 Bloc 4.2.1 ($\lambda = 0$) | 32.05 | 3.13 |
| #4 Bloc 4.2.1 ($\lambda = 1$) | 35.05 | 3.00 |

Les PSNR des images 4 11a au 4 11d varient favorablement de 27 83 à 35 05 dB. Autrement dit, l'algorithme de raffinement proposé par la paire des équations (4 7) et (4 8) procure des améliorations qui sont substantielles et ce, au moins pour l'image de *NewYork City Spin*

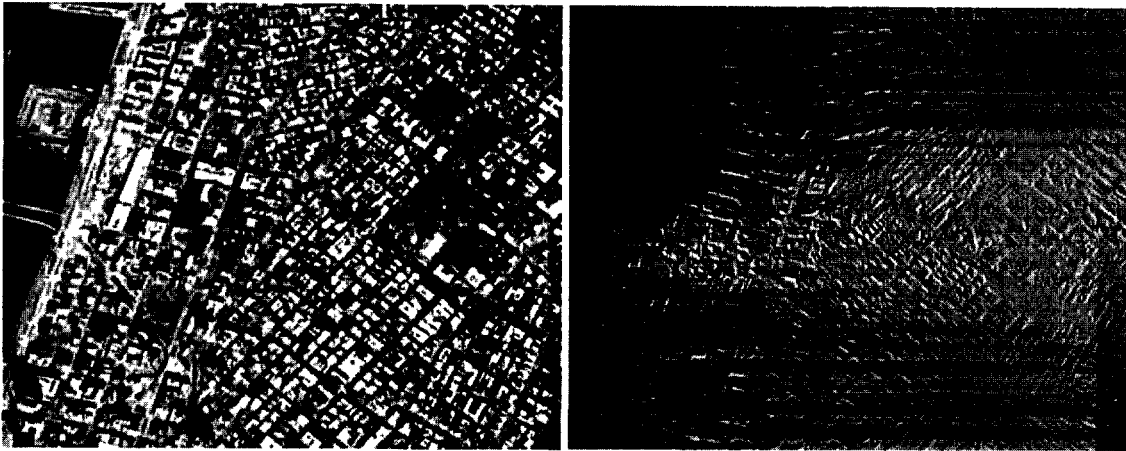


Trame originale

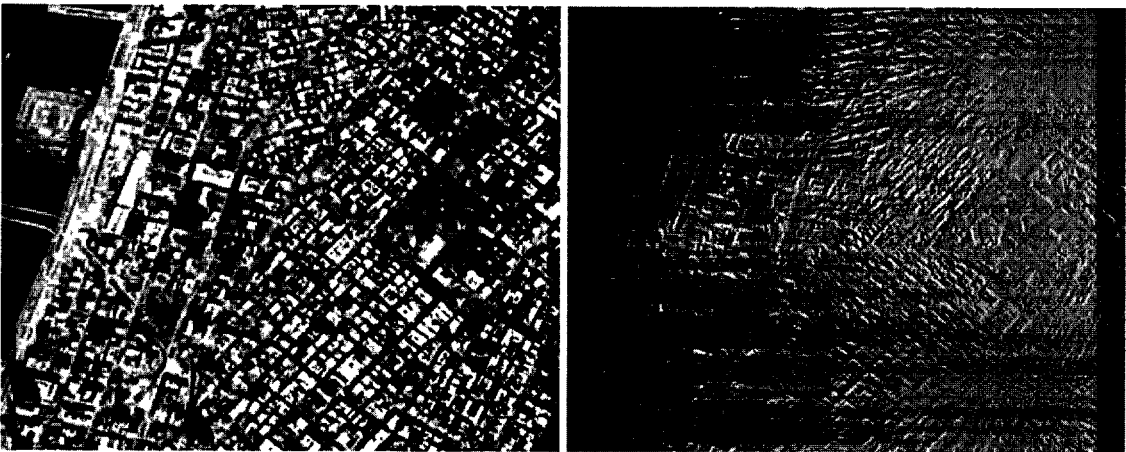


Élargissement de la partie découpée

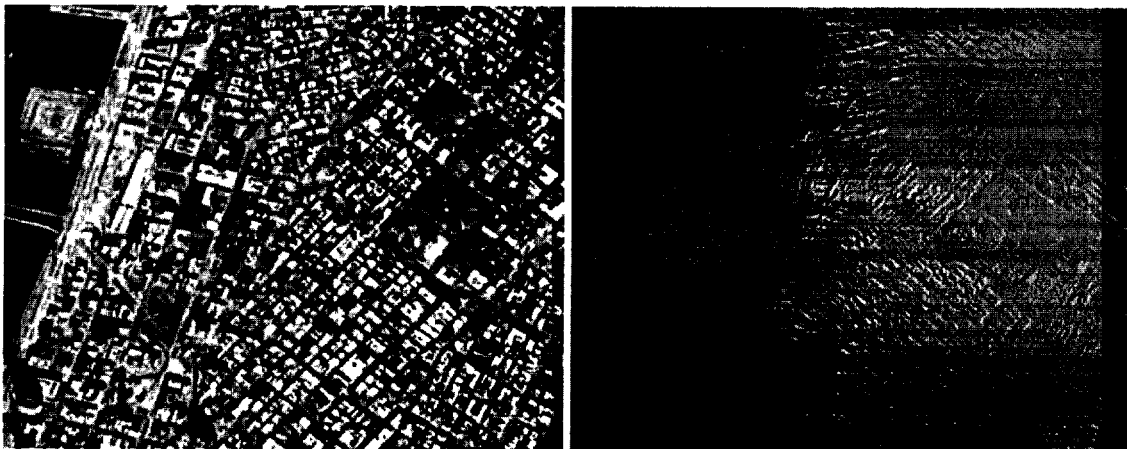
Figure 4 9 Une trame originale de *NewYork City Spin* et la zone d'intérêt découpée



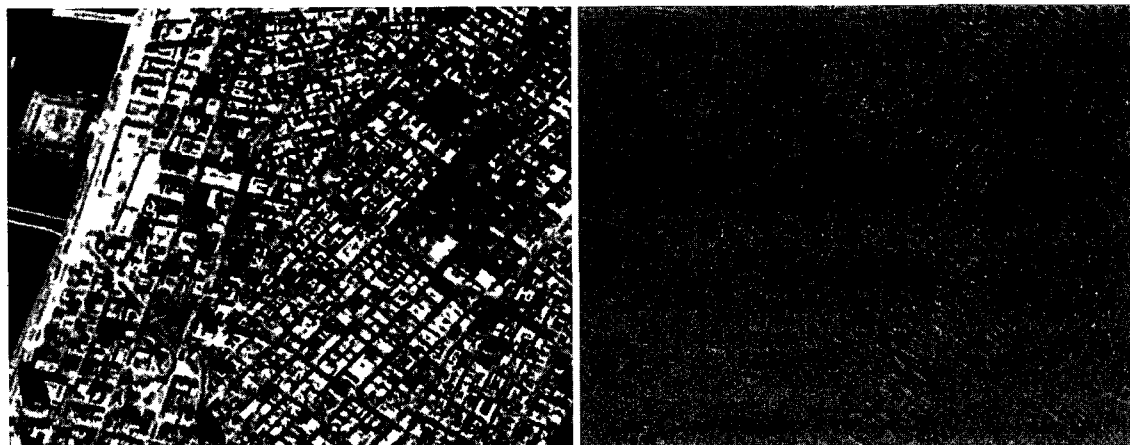
a. Image ré-interpolée découpée et son image d'erreur associée avec vecteurs de mouvement sortis du bloc 4.1.1



b. Image ré-interpolée découpée et son image d'erreur associée avec vecteurs de mouvement sortis du bloc 4.1.2

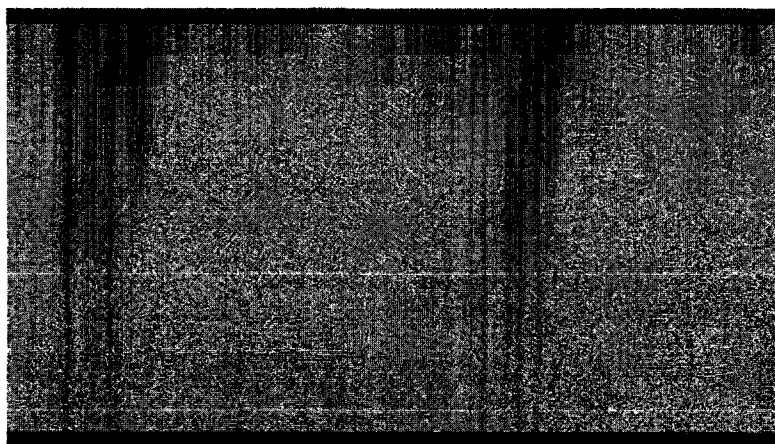


c. Image ré-interpolée découpée et son image d'erreur associée avec vecteurs de mouvement sortis du bloc 4.2.2 et $\lambda = 0$

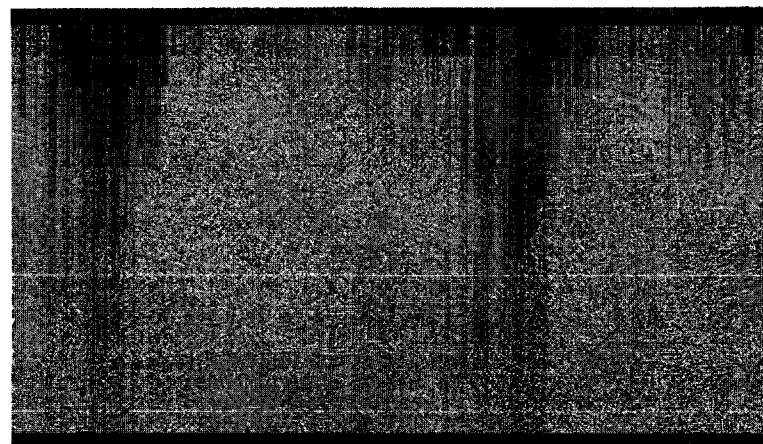


d Image ré-interpolée découpée et son image d'erreur associée
avec vecteurs de mouvement sortis du bloc 4.2.2 et $\lambda = 1$

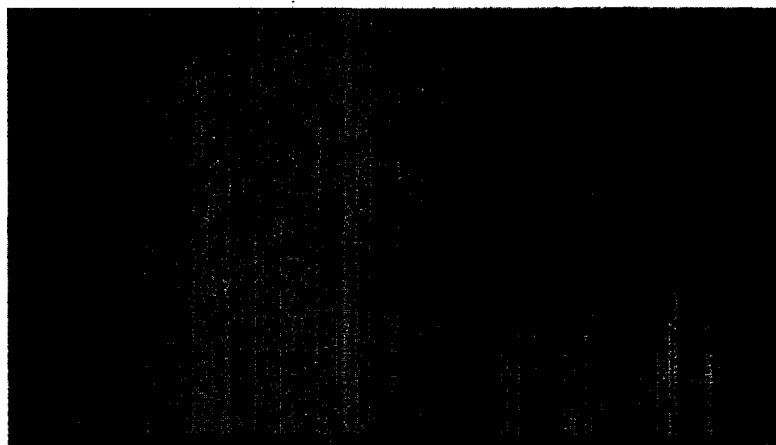
Figure 4 10 Images ré-interpolées et leur image d'erreur respective



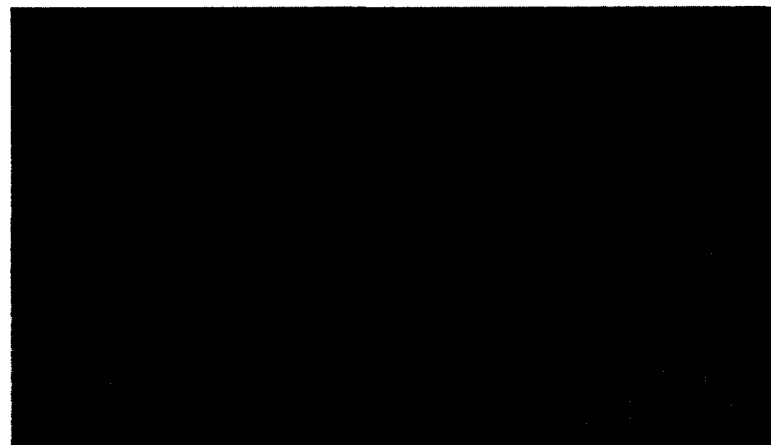
a. Image d'erreur en pleine résolution
avec MVs du bloc 4.1.1 (PSNR = 27.83 dB)



b. Image d'erreur en pleine résolution
avec MVs du bloc 4.1.2 (PSNR = 28.92 dB)



c. Image d'erreur en pleine résolution
avec MVs du bloc 4.2.2 et $\lambda = 0$ (PSNR = 32.05 dB)



d. Image d'erreur en pleine résolution
avec MVs du bloc 4.2.2 et $\lambda = 1$ (PSNR = 35.05 dB)

Figure 4.11. Images d'erreur multipliée par 3 en pleine résolution et PSNR associés

4.6.3 Interpolation générique :

L'interpolation générique proposée à la dernière section 4.5 a un double but

- réduire les artefacts éventuels à l'aide de la pondération en amplitude des images, qui est le coefficient $b(x)$ décrit par les équations (4.17a) et (4.17b)
- créer l'effet de « similitude au film » en rajoutant un peu de saccade tout en réduisant les effets éventuels de scintillement et de halo et ce, à l'aide de d_F et son complément. Ces pondérations des vecteurs de mouvement sont décrites par les équations (4.13) et (4.14)

Ces pondérations, indépendantes entre elles, peuvent être linéaires ou non-linéaires en fonction de la position α , du plan interpolé

En ce qui concerne uniquement la pondération en amplitude, l'interpolation générique proposée est équivalente à l'interpolation non-linéaire proposée décrite par les équations (3.18) – (3.20). Le résultat de la technique a été présenté dans l'article publié. Il ne sera pas nécessaire de le reproduire ici. La section présentera donc les résultats combinant les fonctions non-linéaires en α , à la fois pour les pondérations des images et des vecteurs de mouvement.

Quant à l'aspect de « film similaire » via les pondérations des vecteurs de mouvement, l'effet créé revient au domaine subjectif « plus ou moins artistique » difficile à quantifier objectivement. De plus, l'effet de film similaire à la réduction de saccade, sera mieux visualisé dynamiquement.

Le DVD réservé à cet effet comprend la séquence de *Big Buck Bunny* interpolée par un facteur de 4 et ce, avec les deux types de pondération linéaire et non-linéaire pour fin de comparaison. La figure 4.12 suivante représente les images extraites des séquences interpolées résultantes. Dans la séquence originale, le mouvement du pied du petit lapin et celui du ballon sont les seuls mouvements importants. En dynamique, il y a un certain scintillement et halo autour du ballon si on utilise l'interpolation classique linéaire en position. Ces artefacts sont réduits avec les pondérations non-linéaires. De plus, le petit défaut au pied du petit lapin entre deux images est partiellement corrigé en devenant plus flou.



a Trame originale I_{n-1}



b Trame originale I_n



c Pondération linéaire des vecteurs de mouvement



d Pondération non-linéaire des vecteurs de mouvement

Figure 4 12 Pondérations linéaire et non-linéaire avec la séquence *Big Buck Bunny*

La figure 4 13 suivante extraite de la séquence *Man in Park* illustre la correction possible de la pondération non-linéaire. L'image interpolée est à la position $\alpha = 0.8$.



a Pondération linéaire



b Pondération non-linéaire

Figure 4 13 Pondération linéaire et non-linéaire avec la séquence *Man in Park*

Les résultats de ce chapitre se trouvent dans le répertoire « HD Résultats ». Les séquences à haute définition utilisées (Tableau 4 2) pour démontrer le résultat de ce chapitre se trouvent dans le répertoire « HD originales » du DVD joint avec la thèse à l'annexe C

Tableau 4 2 Séquences HDTV incluses dans le DVD joint avec la thèse

| Nom de séquence | Facteur d'interpolation | Résolution apportée aux MVs | Effet de bloc | Effet du halo | Saccade / Flou | Film Similaire |
|--------------------------|-------------------------|-----------------------------|---------------|---------------|----------------|----------------|
| <i>Walking Girl</i> | 3x | | x | x | | |
| <i>NewYork City Spin</i> | 4x | x | | | | |
| <i>Big Buck Bunny</i> | 4x | | | x | | x |
| <i>Man In Park</i> | 5x | | | x | | x |
| <i>Stairs</i> | 5x | | | | x | |
| <i>Scrolling Text</i> | 3x | | | | x | |

Le tableau complète la description des résultats. Le prochain chapitre sera pour la conclusion.

5 CONCLUSION

Dans ce travail, nous avons proposé plusieurs possibilités pour mettre au point une conversion de taux d'images de performance. Nous concluons en résumant les contributions apportées et en esquissant les extensions possibles de la recherche future.

5.1 Contributions apportées :

La plupart des artefacts créés ou encore les difficultés rencontrées dans les systèmes existants sont dus essentiellement à l'imprécision des vecteurs de mouvement estimés. Dans le contexte d'un système en temps réel, l'imprécision ou encore les erreurs occasionnellement commises dans l'estimation de mouvement, viennent de deux grands facteurs : la complexité des calculs dont l'estimation basée bloc est un exemple classique, et les difficultés rencontrées lors des occlusions qui bloquent les informations requises.

À part des post-traitements pour les cas spécifiques, les contributions proposées dans ce travail comprennent quatre volets : a) la sélection des vecteurs de mouvement à la résolution du pixel (MVS), b) la réduction de halo (HMVR), c) le raffinement hiérarchique de la sélection du vecteur de mouvement basé-pixel à haute résolution et finalement, d) l'interpolation générique d'image.

5.1.1 Sélection des vecteurs de mouvement à la résolution du pixel

Dans le but d'obtenir une estimation de mouvement précise à la résolution du pixel tout en évitant les effets indésirables de trou et de chevauchement des vecteurs, une sélection des vecteurs de mouvement pour chaque pixel a été proposée au chapitre 3.

Dans ce chapitre dédié à la résolution standard, la sélection MVS est effectuée selon un critère pondéré d'erreurs venant à la fois de l'estimation bidirectionnelle et aussi de celle qui est unidimensionnelle. La MVS à chaque pixel est basée simplement sur les vecteurs de mouvement estimés dans les blocs avoisinants. Ceci permet une réalisation possible en temps

réel tout en évitant les techniques itératives proposées dans la littérature. Quant à la qualité de l'image, la technique proposée réduit effectivement l'artefact de bloc.

5.1.2 Réduction de halo HMVR

Lorsque les vecteurs de mouvement sont erronés surtout autour des zones d'occlusion, l'effet de halo crée un artefact très visible et dérangeant. Les erreurs de prédiction, comme SAD, ne sont plus valables. La réduction ou plutôt la correction de cet artefact est difficile, car dans ces zones, l'estimation de mouvement courante ne peut s'appuyer sur des informations valides.

Dans le chapitre 3, nous proposons une nouvelle méthode de correction des vecteurs de mouvement estimés. La méthode consiste en deux étapes : d'abord une interpolation préliminaire est effectuée et ensuite un raffinement des vecteurs de mouvement, pour tout pixel de l'image, est proposé en utilisant les informations obtenues de l'interpolation préliminaire. On évite ainsi une détection explicite des zones de recouvrement ou découverture. Le raffinement, basé sur un modèle de trois régions locales, se fait à l'aide des fenêtres de forme adaptative (SAW) à l'intensité de l'image interpolée préliminaire et aussi aux vecteurs de mouvement de ces trois régions.

La méthode proposée est efficace, la même technique est réutilisée au chapitre 4 pour chaque niveau de la décomposition hiérarchique de l'image.

5.1.3 Raffinement hiérarchique basé pixel du vecteur de mouvement

Au chapitre 4, concernant la résolution à haute définition, l'estimation de mouvement se fait via une décomposition hiérarchique dans le but de réduire la complexité de calcul. Pour obtenir les vecteurs de mouvement basés pixel, un nouveau raffinement est proposé. Le concept de MVS est encore utilisé, mais la généralisation n'est certainement pas triviale ou immédiate. La HD-MVS, pour ainsi l'appeler est basée, sur deux ensembles des vecteurs de mouvement : ceux des neuf (9) pixels avoisinants à la faible résolution et ceux formés du vecteur de mouvement du pixel central plus un faible déplacement fictif dans 8 directions. Le critère de sélection est le minimum des SAD pondérées, dont celles à haute résolution, qui doivent être pénalisées lors de la présence des structures périodiques. La technique combine

ainsi les vecteurs de mouvement estimés de basse résolution et un raffinement à haute définition tout en prévoyant un mécanisme pour éviter les vecteurs de mouvement erronés

5.1.4 Interpolation générique d'image

La première nouvelle interpolation finale proposée est présentée au chapitre 3 via les formules (3-18)-(3-20). Cette contribution quoique modeste permet par contre de produire une image de sortie nette et précise. En fait, l'interpolation finale proposée combine à la fois la position alpha (α) de l'image interpolée, la sélection adaptée à la position alpha des deux images utilisées I_n et I_{n-1} et, la « tendance » locale de fiabilité gamma ($\gamma(x)$) ou son complémentaire ($1-\gamma(x)$) favorisant respectivement le vecteur de mouvement backward ou forward dans une région avoisinante du pixel considéré.

La deuxième interpolation finale plus générique est proposée au chapitre 4, équation (4-18). La proposition consiste en l'introduction des concepts importants dans les pondérations des vecteurs de mouvement et celles des images. D'abord, la distance α de l'image interpolée est révisée avec le concept de la proximité des images existantes. L'image à la distance α sera interpolée comme à une position fictive adéquatement plus approchée de l'une des deux images originales. Cette notion permet le contrôle indépendant des deux types de pondération. De plus, la combinaison pour la première fois des deux paramètres « distance et fiabilité » dans le coefficient de pondération des images rend robuste le résultat de l'interpolation par rapport aux erreurs éventuelles d'estimations de mouvement et leurs artefacts résultants.

Il est important de souligner que les algorithmes proposés sont intégrés dans un ASIC fonctionnant en temps réel, <http://www.hughesnews.ca/1dt-ships-3d-frame-rate-conversion-processors-007804> (page consultée le 22^e juillet 2010) et ce, pour les signaux stéréoscopiques.

Finalement, il est aussi instructif de mentionner que les contraintes et limites du système sont dues essentiellement à la capacité du matériel. La zone de recherche du vecteur de mouvement discutée dans la remarque à la section 4.3 ou la dimension verticale d'une fenêtre sont les exemples typiques. Si l'approche hiérarchique est faisable pour les signaux de télévision à haute ou ultra définition, elle deviendra pratiquement très difficile pour un signal encore à plus

haute résolution. Si c'était le cas, il faudrait recourir, par exemple, au parallélisme du matériel en mettant plusieurs circuits intégrés en parallèle. Chaque circuit intégré travaille sur une portion de l'image.

5.2 Extensions des recherches futures

Dans notre objectif, les extensions possibles des recherches futures mais à court terme peuvent être classées dans deux directions : a) amélioration de la qualité d'image interpolée, b) utilisation des techniques de sélection ou de raffinement des vecteurs de mouvement.

D'une part, pour la qualité de l'image et spécifiquement pour l'effet de halo, l'amélioration du raffinement de la correction des vecteurs de mouvement constituera une avenue possible. En effet, dans ce travail, on n'a utilisé que deux paramètres dans les fenêtres adaptées à la forme. Il s'agit de l'intensité moyenne de l'image préliminaire et les vecteurs de mouvement. On pourrait y ajouter les composantes plus distinctives comme la texture ou les couleurs pour mieux segmenter localement les régions.

D'autre part, si le présent travail permet une certaine précision dans l'estimation des mouvements, on pourra penser à une adaptation de la technique dans la compression temporelle d'inter-trame. Cependant, dans cette optique, il faudrait mettre en balance la réduction de l'erreur de la prédiction versus la transmission ou débit requis pour les vecteurs de mouvement à résolution de pixel.

ANNEXE

Étant donné que la présente étude se réalise dans la majeure partie dans un milieu industriel, plusieurs brevets ont été déposés par l'auteur pour fin de protection de la propriété intellectuelle. Les références les concernant ne sont pas bien connues. Les deux premières annexes ci jointes ont pour but de clarifier la situation pour le lecteur intéressé.

Il s'agira alors des annexes et références suivantes :

- Annexe A Référence [US0161010A1] pour Tran, T H et LeDinh, C -T (juin 2009), *Image Interpolation with Halo Reduction*, U S Patent Application number US 2009/0161010A1
- Annexe B Référence [Tran et al , 2010] pour Tran, T H , LeDinh, P T , Rossignol, F et LeDinh, C T (avril 2010), *High-definition Frame Rate Converter using a new hierarchical approach with halo reduction consideration*, Patent Application Serial No 12/761,214, April 15 2010

Cette référence est volumineuse. Elle sera brièvement présentée.

De plus, dans le but de visualiser les résultats des deux chapitres 3 et 4, les instructions à suivre seront présentées dans l'annexe C.

ANNEXE A – PATENT APPLICATION US 2009/0161010 A1



US 20090161010A1

(19) **United States**

(12) **Patent Application Publication**
TRAN et al.

(10) **Pub. No.: US 2009/0161010 A1**
 (43) **Pub. Date: Jun. 25, 2009**

(54) **IMAGE INTERPOLATION WITH HALO REDUCTION**

Related U.S. Application Data

(75) Inventors **THUY-HA THI TRAN, Hue (VN),
 CHON TAM FEDINH, Montreal (CA)**

(60) Provisional application No. 61/015 228, filed on Dec 20 2007 provisional application No. 61/015 970 filed on Dec 21 2007

Publication Classification

Correspondence Address
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
 901 NEW YORK AVENUE, NW
 WASHINGTON, DC 20001-4413 (US)

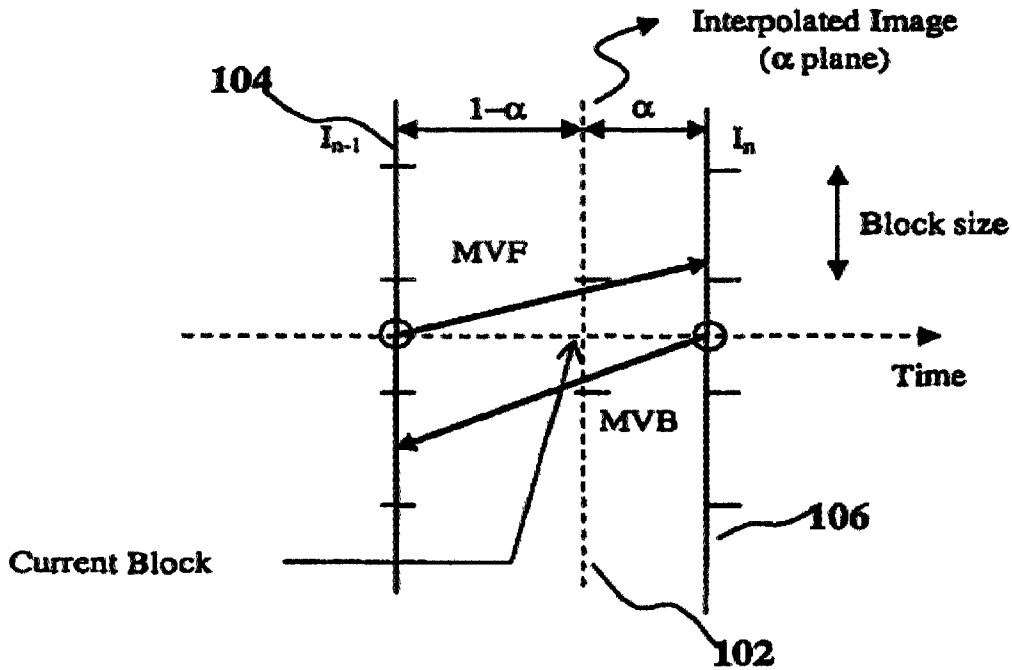
(51) **Int. Cl.** *H04N 7/01* (2006 01)
 (52) **U.S. Cl.** *348/441 348/07 003*
 (57) **ABSTRACT**

(73) Assignee **INTEGRATED DEVICE TECHNOLOGY, INC**

An image interpolator is presented that utilizes a block-based motion estimator to provide block based motion vectors and a motion vector select coupled to the block-based motion estimator to provide pixel based motion vectors. A first image interpolator provides a first interpolated image from the selected motion vectors. A second image interpolator corrects the selected motion vectors, including a correction for halo reduction. Post processing can be utilized to further reduce the halo effect.

(21) Appl No **12/338,954**

(22) Filed **Dec 18, 2008**



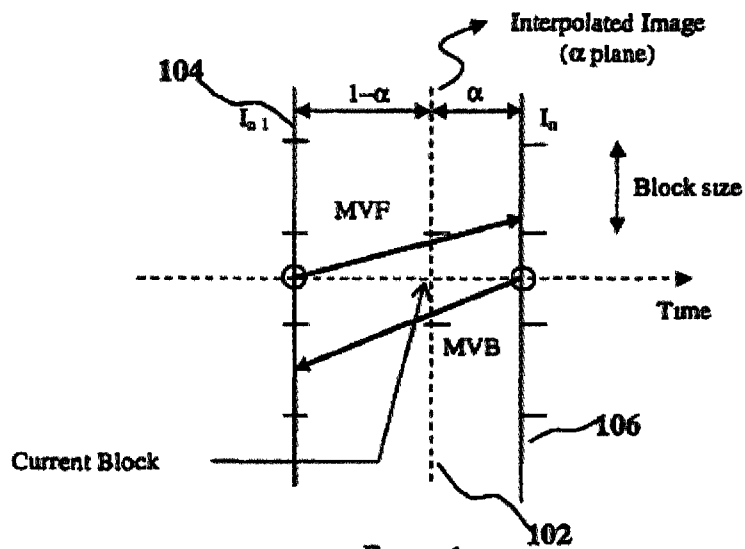


Figure 1

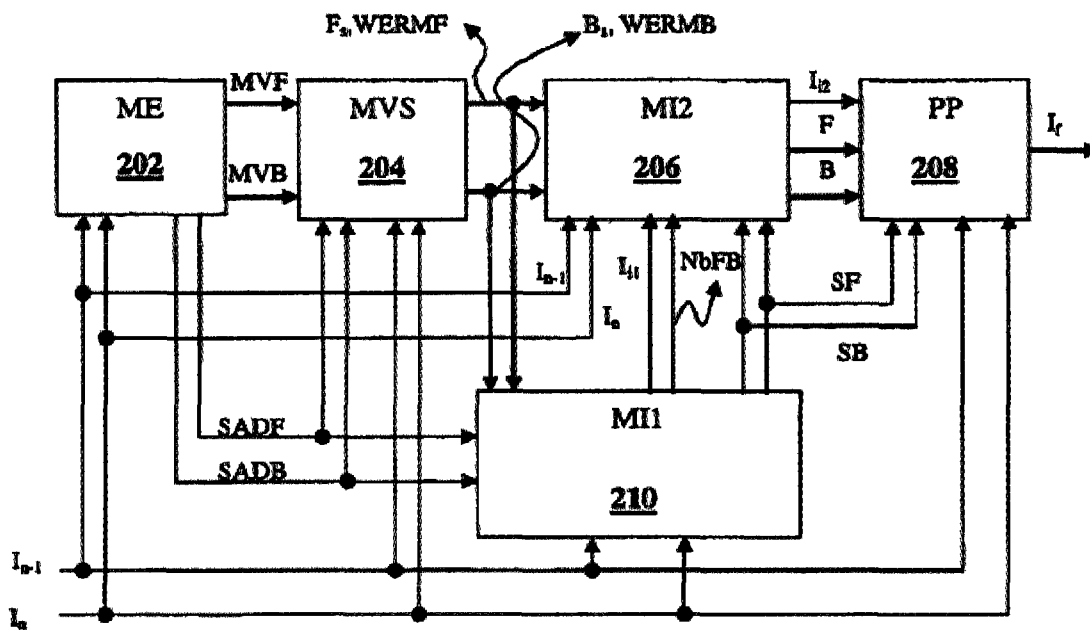


Figure 2

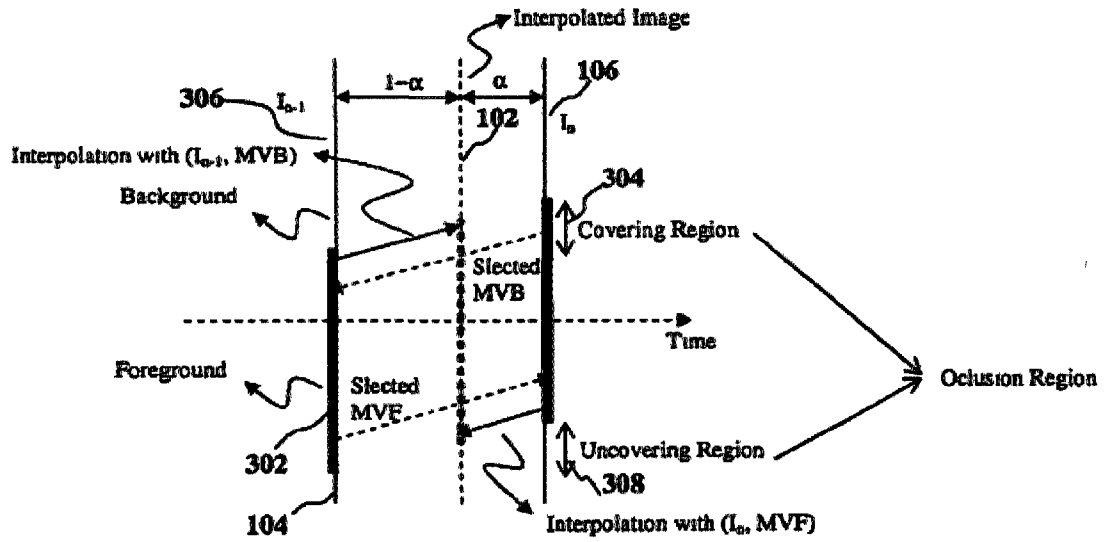


Figure 3

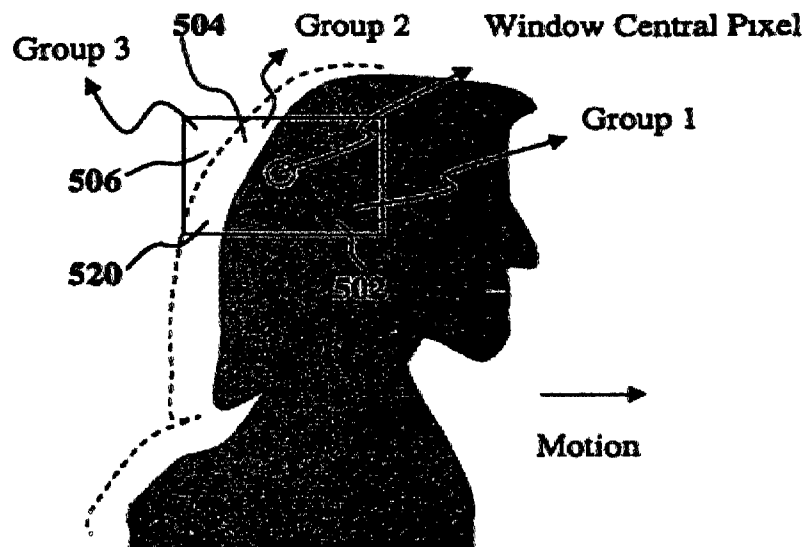


Figure 4A

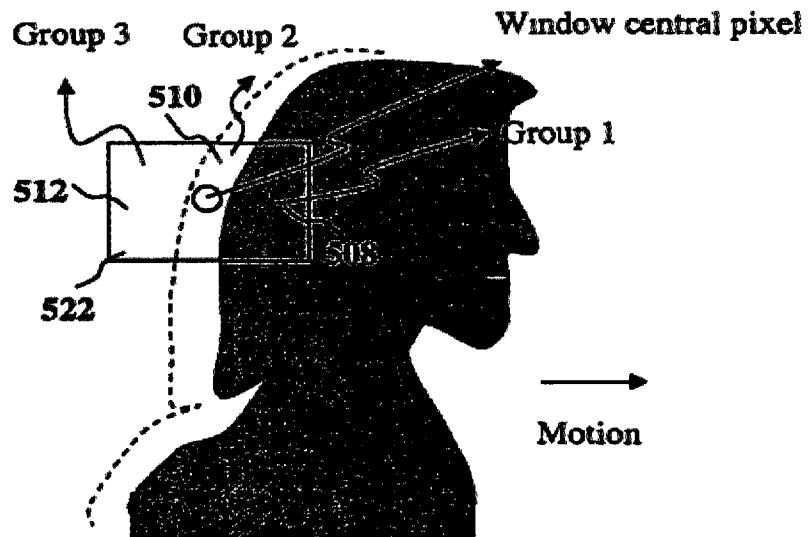


Figure 4B

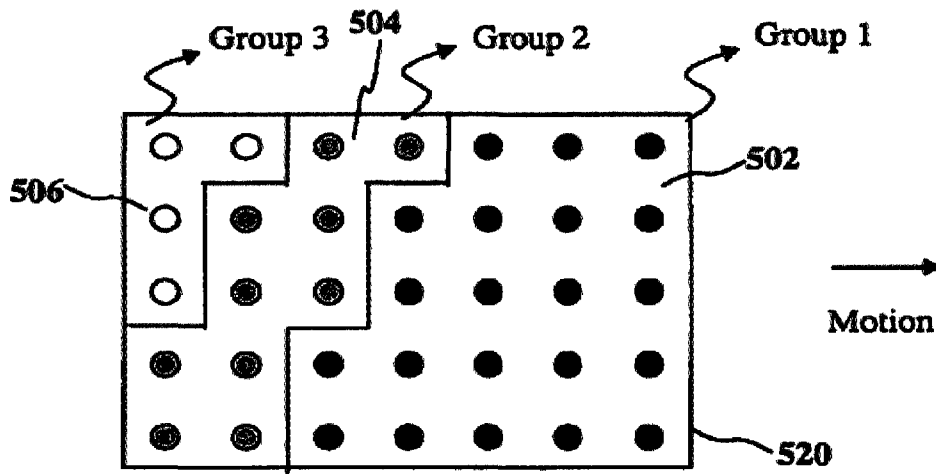


Figure 5A

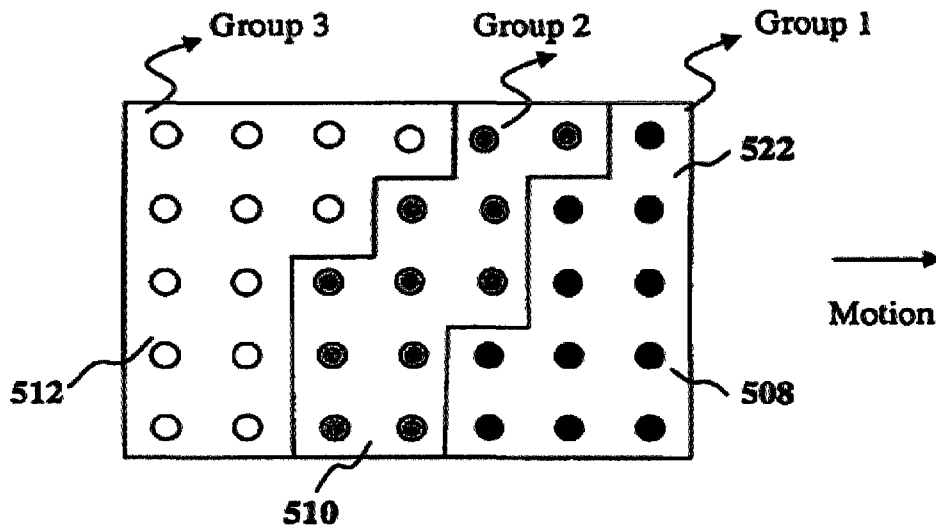


Figure 5B

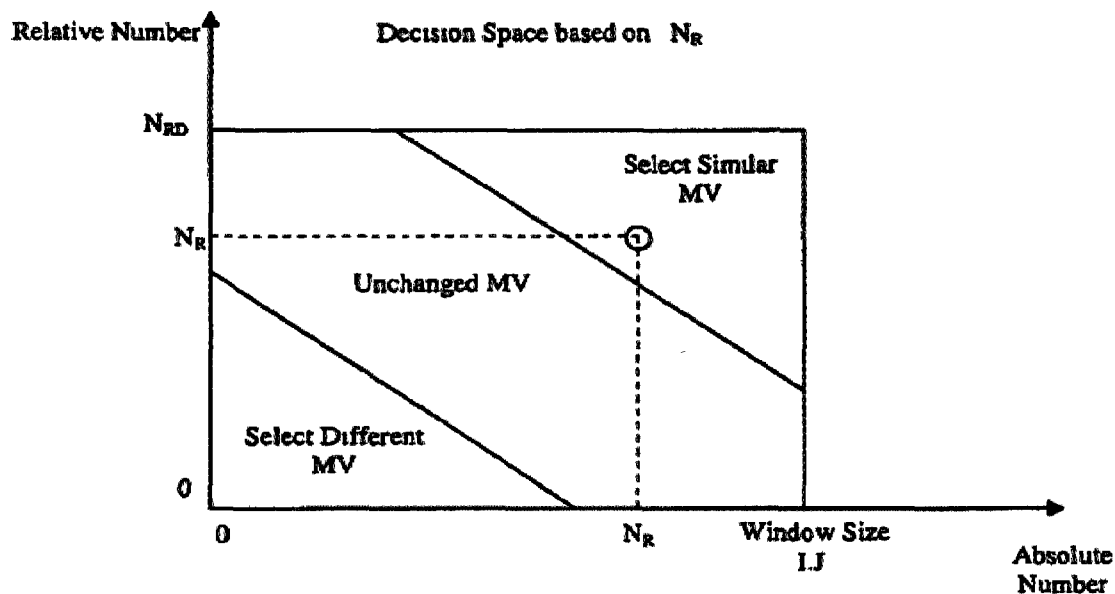


Figure 6

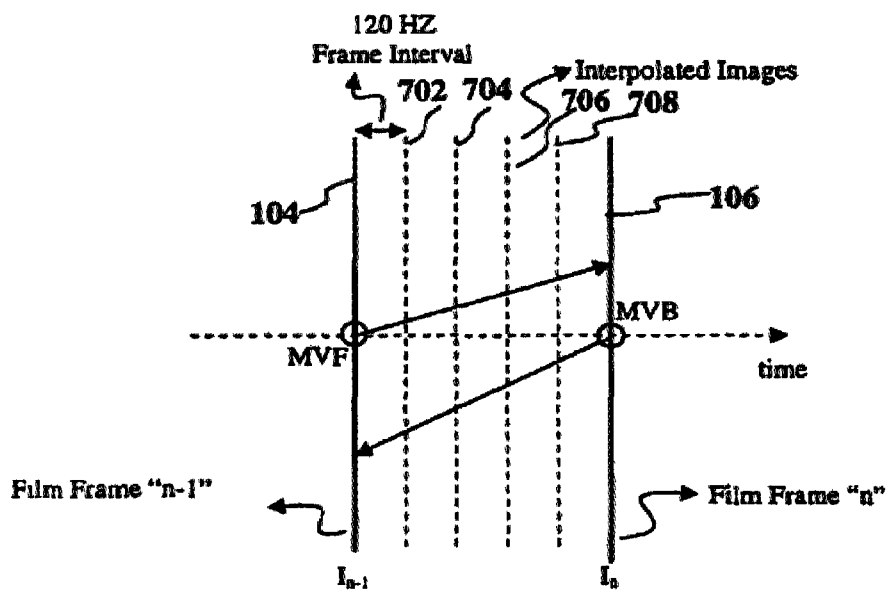


Figure 7

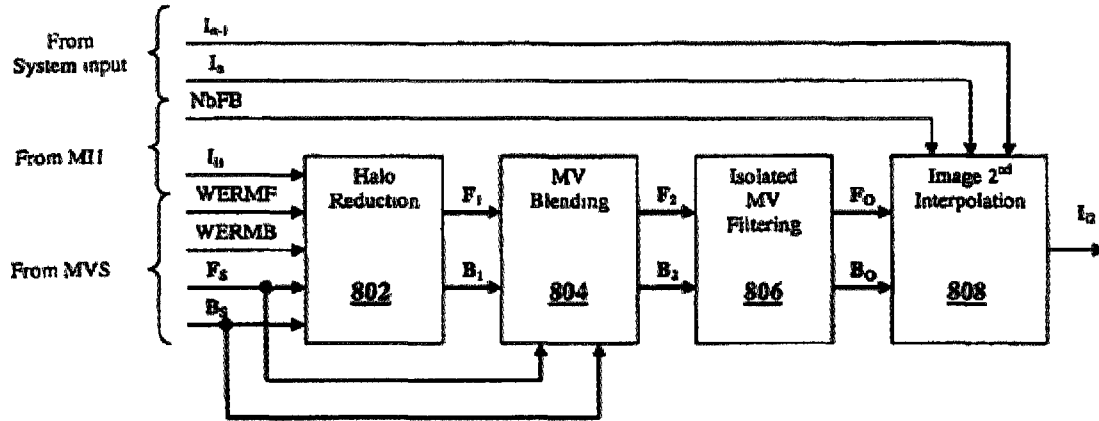


Figure 8

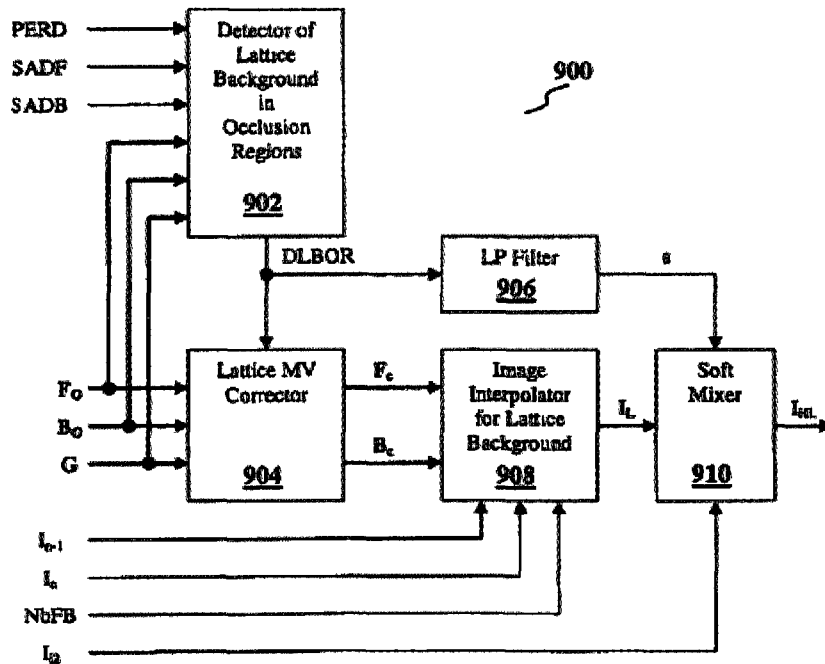


Figure 9

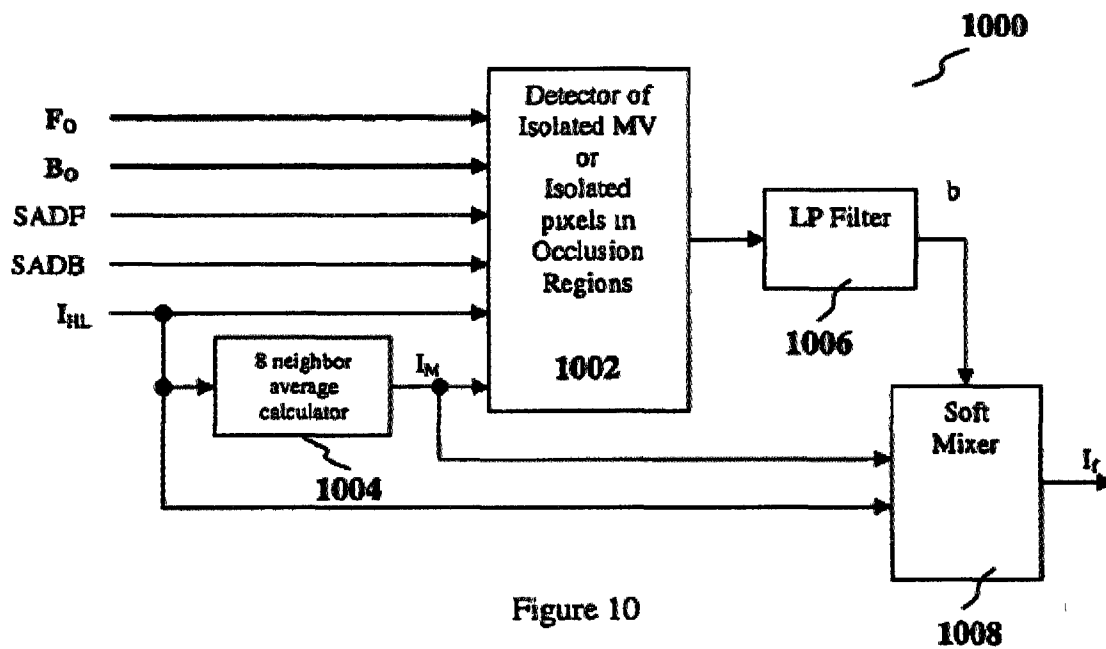


Figure 10

US 2009/0161010 A1

Jun 25, 2009

1

IMAGE INTERPOLATION WITH HALO REDUCTION

PROVISIONAL PRIOR APPLICATION

- [0001] U.S. Patent Application Ser. No. 61/015,228 Dec. 20, 2007
 [0002] U.S. Patent Application Ser. No. 61/015,970 Dec. 21, 2007

US CITATION

References Cited

US Patents

- [0003] U.S. Pat. No. 6,005,639
 [0004] U.S. Pat. No. 6,011,596
 [0005] U.S. Pat. No. 6,219,436 B1
 [0006] U.S. Pat. No. 6,487,313
 [0007] U.S. Pat. No. 6,625,333 B1
 [0008] US 2006/0072790 A1
 [0009] US 2007/0121725 A1
 [0010] U.S. Pat. No. 7,039,109 B2
 [0011] U.S. Pat. No. 7,058,227 B2

OTHER PUBLICATIONS

- [0012] Cr. de Haan "IC for Motion-Compensated De-Interlacing, Noise Reduction and Picture Rate Conversion" IEE Trans. on CE, Vol. 45, No. 3, August 1999
 [0013] K. Sugiyama, I. Aoki & S. Hangan "Motion Compensated Frame Rate Conversion Using Normalized Motion Estimation", 2005 IEEE Workshop on Signal Processing Systems: Design and Implementation (SiPS 2005), Athens, Greece

RELATED APPLICATIONS

- [0014] This application claims priority to U.S. Provisional Application Ser. No. 61/015,228, filed on Dec. 20, 2007, and to U.S. Provisional Application Ser. No. 61/015,970 filed on Dec. 21, 2007, both of which are herein incorporated by reference in their entirety.

BACKGROUND

- [0015] 1. Field of the Invention
 [0016] The present invention relates to reduction of motion compensation artifacts in an image interpolation more specifically to halo reduction in interpolated images.
 [0017] 2. Discussion of Related Art
 [0018] Image interpolation based on motion compensation is a well-established technique. One example of motion compensation is frame-rate conversion (FRC) which is often utilized to increase (or decrease) the refresh rate in video and to have more images (or fewer images) shown per second. One advantage of interpolating video to a higher number of images per second is that, with the higher refresh rates available in LCD panels, motion appears more fluid. Another advantage is in the conversion of film content which is normally at 24 frames per second (fps) to video, which could be 30, 50, 60, 100, 120, 200 or 240 fps. An advantage of interpolating to fewer images is in conversion between commonly used standards such as 60 Hz for National Television System Committee (NTSC) to 50 fps for the Phase Alternating Line (PAL) standard.

[0019] FRC is very challenging, particularly in real-time video. Many previous solutions have attempted to address this issue, however these solutions all introduce artifacts, especially in real-time applications. Most prior art methods use block-based motion vector (MV) estimation for motion compensation (MC) and motion estimation (ME). These prior art solutions, even for progressive images, cause many artifacts, including the halo effect, flickering, and block artifacts.

[0020] Block artifacts can occur when the MV estimation is based on a block of pixels of fixed and predetermined size. In this case, all pixels in a block are supposed to have the same MV even if within a block there are different objects or regions moving in different manners. Moreover, when adjacent blocks to a given block have different estimated motion vectors for any reason, the block boundaries become more obvious as same-region pixels will be displaced differently in different adjacent blocks.

[0021] Flickering is another artifact in motion compensated image interpolation. This artifact occurs when the motion vectors of a few pixels in the image sequence are not consistent from one frame to another frame. The inconsistency is generally due to erroneous motion vector calculations in a small region.

[0022] Halo effects occur around a foreground object in an image in a video sequence when the object is moving relative to a background. The background could be stationary and the object moving or vice versa. In some cases, both the object and the background are moving, as may be the case for example, in a video game sequence. The origin of the halo effect is erroneous MV estimation in occlusion areas. The visual perception of the halo effect is that objects seem to have a halo of different characteristic around them. For example, the face of a person walking in a scene could seem to be surrounded by a halo, as if a portion of the background was moving along with the face.

[0023] There have been various halo reduction algorithms developed. Halo reduction can be based on Object-based Interpolation as described in U.S. Pat. No. 6,625,333. However, this technique implies object segmentation and object coloring which requires additional frame memory and additional bandwidth. Moreover, the proposed MV correction requires multiple frame-based iterations which are not suitable for real-time processing.

[0024] Multi-frame solutions, those that involve analysis of more than two frames of data, have also been proposed. See, e.g., U.S. Pat. No. 6,005,639; U.S. Pat. No. 6,011,596; U.S. Pat. No. 7,010,039; U.S. Application Publication No. 2007/0121725 A1 and K. Sugiyama, I. Aoki & S. Hangan "Motion Compensated Frame Rate Conversion Using Normalized Motion Estimation" 2005 IEEE Workshop on Signal Processing Systems: Design and Implementation (SiPS 2005), Athens, Greece. Multi-frame solutions for covering/uncovering/no occlusion detection and MV correction are based essentially from previous-previous and far-future images frames. Multi-frame solutions require large image memory and highly increased bandwidth between the memory and the processor.

[0025] Two-frame solutions that have been proposed are based on MV correction (or re-estimation), interpolation strategies, and/or a mixing of these two methods. In MV correction, separating foreground and background motion vectors is also a possible way to reduce the halo effects. The use of some motion vectors set at borders of the image as possible background MV has been proposed in U.S. Publica-

tion No. US 2006/0072790 A1. The assumption that the border MV is the background MV, however, is not necessarily valid and the background MV estimation requires additional frame latency and extra memory.

[0026] Median filtering, an example of interpolation strategy, is proposed in G. de Haan, "IC for Motion-Compensated De-Interlacing, Noise Reduction and Picture Rate Conversion", IJELT Trans. on CI, Vol. 45 No. 3, August 1999. In this reference, Median Filtering Interpolation is the median value of forward image interpolation, backward image interpolation, and the averaging value of the two existing images. Moreover, forward and backward MV are obtained using a single recursively estimated MV. Median filtering interpolation, as described, can yield generally good results in non-occlusion regions. However, in occlusion regions the estimated MV utilizing the Median filtering method is no longer correct.

[0027] A weighted mean of individual interpolations is proposed in U.S. Pat. No. 7,039,109. Each individual interpolation provided by the MV in a neighbor group is based again on Median filtering of forward and backward interpolations and the mean value of the two existing images. MVs described in this patent are block-based and forward estimated from the present image to the next image. The weight specified for an individual interpolation is a measure of reliability in function of the difference of a predicted luminous intensity and/or of the relative frequency of occurrence of the MV in the neighboring groups. MV edge detection is also provided for complexity reduction purposes. However, the group or block-based MV solution can provide blocking effects. Additionally, a single (forward) MV for a block can be insufficient in occlusion parts of the picture. Further MV edge detection is only good for software simulation. Also, pixel-based image averaging can blur the interpolated image in the occlusion regions or yield an image doubling effect when the MV is large.

[0028] Covering/Uncovering Detection, a second block-based MV with error at 2^{nd} or 4^{th} power, is proposed for more precision in U.S. Pat. No. 6,219,436. MV correction by using MV at shifted locations in the function of detected "real" covering/uncovering regions is proposed. However, various median filtering strategies with appropriate mean values for detected regions are in fact the preferable solution presented in the patent.

[0029] In both U.S. Pat. No. 6,487,313 and U.S. Pat. No. 7,058,227, it is assumed that the discontinuities in MV field correspond to the borders of moving objects. The use of MV length to determine the occlusion areas, an additional bidirectional interpolation error calculation with small block size for increasing the precision of detected occlusion areas, and various median filtering for final interpolation is thus proposed. However, MV length is not necessarily a reliable parameter when the objects are composed of flat but noisy regions. The discontinuities in MV field and borders of moving objects require some expensive precision, and median filtering is still an ad-hoc technique since MV is not corrected.

[0030] Therefore, there is a need to provide a fast interpolation of image planes that does not require large amounts of memory to accomplish.

SUMMARY

[0031] An image interpolator that is consistent with embodiments of the present invention can include a motion estimator that estimates block-based motion vectors between

a first image and a second image, a motion vector select coupled to the motion estimator that provides selected motion vectors based on the block-based motion vectors, a first interpolator coupled to receive the selected motion vectors and provide a first interpolated image for a plane between the first image and the second image, and a second interpolator coupled to receive the first interpolated image that corrects the selected motion vectors to form corrected motion vectors and provides a second interpolated image based on corrected motion vectors.

[0032] Another image interpolator that is consistent with the present invention may include a block-based motion estimator coupled to receive adjacent images, the block-based motion estimator providing a block-based motion vector, a block-based motion vector forward error, and a backward error, a motion vector select coupled to the block-based motion estimator, the motion vector select providing pixel based selected forward motion vector and selected backward motion vector, and providing pixel based forward and backward errors, a first image interpolator coupled to the block-based motion estimator and the motion vector select, the first image interpolator providing a first interpolated image from the selected forward motion vector and the selected backward motion vector, a second image interpolator coupled to the first image interpolator and the motion vector select, the second image interpolator providing a second interpolated image and corrected forward motion vectors and corrected backward motion vectors, and a post processing block coupled to the first image interpolator and the second image interpolator, the post processing block providing a final interpolated image from the second interpolated image and the corrected forward motion vectors and the corrected backward motion vectors.

[0033] A method for interpolating an image between first and second adjacent images consistent with embodiments of the present invention includes estimating a block-based forward motion vector and a block-based backward motion vector between the first and second adjacent images, selecting a forward motion vector for each pixel in the image based on the block-based forward motion vector and a forward error function, selecting a backward motion vector for each pixel in the image based on the block-based backward motion vector and a backward error function, interpolating a first image from the selected backward motion vectors, the selected forward motion vectors, a block-based forward estimation error and a block-based backward estimation error; interpolating a second image from the first image based on forward and backward weighted errors of the selected forward motion vector and the selected backward motion vector, and determining the image from the second image.

[0034] These and other embodiments consistent with the invention are further described below with reference to the following figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0035] FIG. 1 illustrates interpolation of an image plane between two adjacent images.

[0036] FIG. 2 shows a block diagram of an embodiment of an image interpolator consistent with the present invention.

[0037] FIG. 3 illustrates foreground and background and covering and uncovering regions utilized in some embodiments consistent with the present invention.

[0038] FIG. 4A illustrates a moving object and a sliding window centered in a foreground region.

US 2009/0161010 A1

3

Jun 25, 2009

[0039] FIG. 4B illustrates a moving object and a sliding window centered in an occlusion region.

[0040] FIG. 5A illustrates a sliding and three-region segmented window centered in the foreground region.

[0041] FIG. 5B illustrates a sliding and three-region segmented window centered in the occlusion region.

[0042] FIG. 6 illustrates a possible decision for MV correction.

[0043] FIG. 7 illustrates interpolating multiple images between two existing consecutive images.

[0044] FIG. 8 illustrates a second motion interpolator as shown in FIG. 2 for halo reduction.

[0045] FIG. 9 illustrates further processing for halo reduction in background lattice post-processing.

[0046] FIG. 10 illustrates processing for flicker reduction that can be performed in post-processing.

[0047] In the figures, to the extent possible, elements having the same or similar functions have the same designations.

DETAILED DESCRIPTION

[0048] In a frame rate conversion (FRC) application, the "halo effect" is a visual artifact that occurs when a foreground object is moving against a detailed background. A system for reducing the halo effect consistently with the present invention includes a pixel-based motion vector (MV) selection, a preliminary interpolation, local shape adaptive windowing for MV foreground/background correction, and a final interpolation with some additional post-processing. The preliminary and final image interpolations are based on occurrence frequency of local Sum of Absolute Differences (SADs) of the forward and backward motion vector estimations. The MV correction for halo reduction can then be based substantially on local image intensities and on local MV variations. Some embodiments are non-iterative techniques that are efficient for image memory usage and communication bandwidth between the image memory and the image processor.

[0049] FIG. 1 illustrates the time position of an interpolated image (α -plane) 102 to be interpolated from two consecutive images 104 and 106. (Contrary to proposed multi-frame solutions (See US Pat No. 6,005,639, US Pat No. 6,011,396, US Pat No. 7,010,039, US Publication No. 2007/0121725A1, and K. Sugiyama, I. Aoki & S. Hangar "Motion Compensated Frame Rate Conversion Using Normalized Motion Estimation") interpolated image 102 is based on information from only two consecutive images: image 104 I_n and image 106 I_{n-1} . Images 104 and 106 are used to interpolate image 102 at an arbitrary temporal distance referred to as the "Alpha (α) plane" located a distance α in time from image 104 I_n and a distance $(1-\alpha)$ from image 106 I_{n-1} , as shown in FIG. 1. Because only two existing images are used for the interpolation, some embodiments consistent with the present invention are advantageous in terms of memory bandwidth, and speed. Some embodiments of the present invention yield visually desirable interpolated images and provide acceptable real-time FRC solutions.

[0050] FIG. 2 illustrates an embodiment of an image interpolator 200 consistent with the present invention. Data representing images 104 and 106, I_n and I_{n-1} , are input to image interpolator 200. An interpolated image, I_α , corresponding to the image in α -plane 102, is output from image interpolator 200. Image interpolator 200 includes a motion estimator (ME) 202, a motion vector selector (MVS) 204, a first motion interpolator (MI1) 210, a second motion interpolator (MI2)

206 and a post processing (PP) block 208. The image inputs I_{n-1} and I_n are received by each of MI 202, MVS 204, MI1 210, MI2 206, and PP 208.

[0051] Motion Estimator 202 analyzes image data I_n and I_{n-1} and provides forward motion vectors (MVF) and backward motion vectors (MVB), as shown in FIG. 1 for a single block. ME 202 may be a block matching algorithm used to estimate forward motion vector MVF and backward motion vector MVB. Forward motion vector (MVF) indicates the displacement of a given image block in image 104 I_n with respect to a matched block in image 106 I_{n-1} . Backward motion vector (MVB) indicates the displacement of a given block in image 106 I_{n-1} with respect to a matched block in image 104 I_n . In addition, ME 202 generates a forward and backward Sum of Absolute Differences (SAD) and SADB for the full search estimated to produce MVF and MVB, respectively. Mathematically SAD, SADB, MVF, and MVB can be defined as follows:

$$SAD = \min_k \left(\left| \sum_{(x,y) \in BW} |I_n(x+y) - I_{n-1}(x+k)| \right| / N(BW) \right) \quad (1)$$

$$SADB = \min_k \left(\left| \sum_{(x,y) \in BW} |I_{n-1}(x+y+k) - I_n(x+y)| \right| / N(BW) \right)$$

$$MVF = k + \text{minimizing SAD in a given search zone}$$

$$MVB = k + \text{minimizing SADB in a given search zone}$$

In the equations, $N(BW)$ denotes the size of the block BW , x and y are pixel coordinates of the current block BW and, x and y are the coordinates of pixels within the current block BW . For precision purposes, the SAD, SADB and the corresponding motion vectors can be associated with a small window block B of pixel size $m \times n$ located inside block BW . A block-based MV field is therefore generated as a result for each of the images 104, I_n , and 106, I_{n-1} . An example of a motion estimator that can be utilized as motion estimator 202 is further described in US Pat App. [Attorney Docket No. 09145-0091-00000], filed contemporaneously with the present disclosure, which is herein incorporated by reference in its entirety.

[0052] In some embodiments, the motion estimation performed in ME 202 can be adaptive for a particular search range, with or without lattice detection or estimation. ME 202 with lattice detection can also provide a block-based periodic detected signal and a global MV, when it is available.

[0053] In embodiments appropriate for interlaced images, the motion estimation performed in ME 202 can be performed directly on adjacent fields with same and/or opposite parity. Alternatively, in some embodiments adapted to minimize eventual aliasing effects, motion estimation can be performed on de-interlaced image versions with suitable filtering and cadence detection.

[0054] MV Selector (MVS) 204 converts the MV field (i.e., all the MVFs and MVBs from ME 202) from block-based to pixel-based in the alpha (α) plane of image 102. As such, MVS 204 is coupled to receive the forward and backward motion vectors MVF and MVB and the values SAD and SADB from ME 202 as well as image data I_{n-1} and I_n . Conversion from block-based data to pixel-based data, in some embodiments, is effective in reducing the blocking effect. MVS 204 provides each pixel with its own forward MV and

its own backward MV and generates two pixel-based MV fields corresponding to forward and backward motion vectors [0055] MVS 204 takes the fields MVI and MVB and performs pixel-based motion vector selection for each of MVI and MVB in the α -plane of image 102. For each temporal and spatial pixel position P in the α -plane, a $m \times n$ pixel block surrounding pixel position P is considered, where P is significantly central to the $m \times n$ block of pixels. Next, areas in both images 104, I_n , and 106, I_{n-1} , consisting of a number $p \times q$ of equivalent blocks of $m \times n$ pixels around the positions corresponding to P in images 104 and 106 are examined, which amounts to $p \times q \times m \times n$ block-of-pixels areas. These areas in data I_n of image 104 and data I_{n-1} of image 106 are therefore $p \times q$ times larger than the $m \times n$ block in the α -plane 102. Two $m \times n$ pixel blocks in the two image planes 104, I_n , and 106, I_{n-1} , which correspond to the spatial position of the current $m \times n$ pixel block in α -plane 102, as illustrated by FIG. 1, are significantly central to these larger areas respectively. All the block-based MVs that are the output from ME 202 and are pointing from plane 104, I_n , to plane 106, I_{n-1} , in the neighborhood defined by the $p \times q \times m \times n$ pixels. These selected MVI and MVB results, I_f and B_f , are output from MVS 204.

[0056] The horizontal Neighbor Block Number p, for example, could be determined as a function of the horizontal component length of motion vector MVI, the α value, and the horizontal block size m. However, in some embodiments, a fixed square ($\pm 2 \times \pm 2$) ($p = 5$ and $q = 5$) of neighbor vectors around pixel x can be considered for all α .

[0057] At a considered pixel of coordinates x, for each of 25 MVI candidates, a local error LFRF_{pq}(X) can be calculated in a given window w of N(w) pixels as follows. LFRF_{pq}(X) can be given by

$$LLRF_{pq}(x) = \sum_{i=k}^{i=k+\lambda} |I_n(x+k+\lambda+i) - \alpha MVB_{pq}(x) - I_{n-1}(x+k-i) - (1-\alpha)MVI_{pq}(x)| \cdot N(w) \quad (2)$$

Because λ , k, and MVI_{pq} are integers and α is fractional, the required values $I_n(x+k+\alpha MVI_{pq})$ and $I_{n-1}(x+k-(1-\alpha)MVI_{pq})$ in Eq. (2) can be obtained by using appropriate spatial interpolation of the two images.

[0058] In some embodiments, a composite local error CERF_{pq}(x) defined as below yields better results.

$$CERF_{pq}(x) = \alpha LFRF_{pq}(x) + (1-\alpha)SADF_{pq} \quad (3)$$

in which the parameter α is greater than 1/2.

[0059] In order to protect the smoothness in the MV field, a penalty factor pf and a penalty offset po are introduced in the following weighted local error WERF_{pq}(x)

$$WERF_{pq}(x) = pf(CERF_{pq}(x)) + po \quad (4)$$

In some embodiments, for example, pf can be set to 2 and po can be set to either 1 or 2 when the horizontal or vertical component of the motion vector MVI differs from a corresponding neighbor mean value by a given threshold. Otherwise, pf can be set to 1 and po set to 0. In the presence of halftone at the current pixel location, the threshold value can be set to be smaller.

[0060] The selected forward and backward motion vector I_f is the one that minimizes the error function. For example,

if the weighted error function is utilized the selected forward motion vector I_f corresponds to the MVI_{pq} that minimizes $WERF_{pq}(X)$. Moreover, in order to avoid some conflict situations when some $WERF_{pq}(X)$ are equal for some values of p and q, certain block priorities P_{pq} can be defined. The priority can be a function of distance from the central block, which has the highest priority.

[0061] The selected backward motion vector B_f can be similarly chosen. The associate local error LFRB_{pq}(x), the composite error CERB(x) and the weighted error WERB(x) are given respectively by

$$LFRB_{pq}(x) = \sum_{i=k}^{i=k+\lambda} |I_{n-1}(x+k-i) - \alpha MVB_{pq}(x) - I_n(x+k+i) - (1-\alpha)MVI_{pq}(x)| \cdot N(w) \quad (5)$$

$$CERB_{pq}(x) = \alpha LLRB_{pq}(x) + (1-\alpha)SADB_{pq} \quad (6)$$

$$WERB_{pq}(x) = pf(CERB_{pq}(x)) + po \quad (7)$$

[0062] Thus, at each pixel x in the α -plane 102, MVS 204 outputs a pair of forward and backward motion vectors I_f and B_f and their respective associate optimum weighted local error WFRMf and WERMB. The results of the calculations performed in MVS 204 can then be utilized in First and Second Motion Interpolators, MI1 210 and MI2 206.

[0063] In some embodiments, MI1 210 and MI2 206 provide Halo effect reduction and image interpolation. Contrary to solutions previously described (e.g. in US Pat. No. 6,219,436 (which teaches utilization of a second motion estimator) and US Pat. No. 6,487,313 or US Pat. No. 7,058,227 (which teach utilizing a MV length)), the occlusion covering and uncovering regions are not explicitly detected in MI1 210 and MI2 206.

[0064] In some embodiments, First Motion Interpolator (MI1) 210 operates "on-the-fly" or in a pipeline. MI1 210 may not utilize the entire interpolated image for processing in MI2 206. MI1 produces a preliminary interpolated image in α -plane 102 from original images 104, I_n , and 106, I_{n-1} . The preliminary interpolation further processed in MI1 210 may also utilize the selected MVI, I_f , selected MVB, B_f , local SADF, and local SADB. In principle, one of the selected MVI and MVB is used in the occlusion covering and uncovering areas.

[0065] Covering and occluded areas are illustrated in FIG. 3. As shown in FIG. 3 for covering region 304, the intersection of the MVs and α plane 102 corresponds to background region 306 in I_n , and to foreground region 302 in I_{n-1} . Accordingly, the selected MVB, B_f , and I_{n-1} are used for this region. This procedure can minimize the probability of the selected MVI, I_f , pointing from a background object to a foreground object and vice versa, as shown in FIG. 3. Similarly, for uncovering regions 308, the selected MVI, I_f , and I_n are used. However, in the occlusion regions which is the union of covering and uncovering regions 304 and 308, neither motion vectors I_f or B_f are valid. Moreover, in order to provide more robustness, a context-based interpolation can be used by combining (I_n, I_f) and (I_{n-1}, B_f). The context is based on the pixel number NbFB in a $M \times N$ sized sliding window, NbFB is in turn, the number of pixels which should to be interpolated with B_f , the selected MVB.

[0066] Let SF and SB be pixel-based values of SADF and SADB respectively. Further let S1 and S2 be appropriate

threshold values. The value for the number of pixels $NbFB$ can be defined as a number of pixels in the sliding window $M \times N$ such that

[0067] If α is greater than or equal to $1/2$, then

$$NbFB = \# \text{ of pixels } (SB < S1) \text{ or } (SB - S1) < SF + SF + S2 \quad (8)$$

In other words, $NbFB$ is the number of pixels such that their SB is smaller than the threshold $S1$ or their SB is greater or equal to the threshold $S1$ and smaller than the sum of their $S1$ and a second threshold value $S2$.

[0068] If α is smaller than $1/2$, then

$$NbFB = (M \times N) - \# \text{ of pixels } (SF < S1) \text{ or } ((SF - S1) < S2) \quad (9)$$

In other words, $NbFB$ is now the complement of the number of pixels such that their $S1$ is smaller than the threshold $S1$ or their $S1$ is greater or equal to the threshold $S1$ but smaller than the sum of their $S1$ and a second threshold value $S2$. $S1$ and $S2$ can be set equal respectively 5 and 1.

[0069] Let F be the value of $NbFB$ normalized by the window size and $1-F$ the corresponding complementary value. In some embodiments, M11 210 can then perform the context based interpolation

$$I_{11}(x) = EB I_0(x + \alpha F) + FF I_0(x + (1 - \alpha)B), \quad (10)$$

In some embodiments, instead of the interpolation described in Eq. 10, M11 210 can form an interpolation that is a function of temporal distance as the following

$$I_{11}(x) = EB I_0(x + (1 - \alpha)F) + FF I_0(x + (1 - \alpha)B), \quad \text{if } \alpha < 1/2$$

$$I_{11}(x) = EB I_0(x + \alpha F) + FF I_0(x - \alpha B), \quad \text{if } 1/2 < \alpha < 1$$

$$I_{11}(x) = FF I_0(x + \alpha F) + EB I_0(x + (1 - \alpha)B), \quad \text{if } 1/2 < \alpha < 1 \quad (11)$$

M11 210 then outputs the interpolated image I_{11} , the local pixel number $NbFB$, the local pixel-based $S1$ and the local pixel based SB to M12 206 which performs halo reduction and a second image interpolation.

[0070] Second Motion Interpolator (M12) 206 utilizes the results produced by M11 204 as well as those produced by MVS 204 with shape-adaptive windowing in α -plane 102 for a second interpolation of the α -plane 102. FIG. 8 illustrates an embodiment of M12 206 consistent with the present invention. As shown in FIG. 8, in some embodiments, M12 206 includes a Halo reduction block 802, a MV blending block 804, an isolated MV filtering block 806, and an image second interpolation block 808.

[0071] In some embodiments, Halo reduction 802 uses a local sliding window through the first interpolated image I_{11} and the MV field F_x and F_y in α plane 102 to filter the MV and thereby generate the α plane image 102. Explicit detection of covering regions 304 or uncovering regions 308 need not be utilized. The local sliding window consists of a block of $[x, y]$ pixels and sweeps across and down α -plane 102, one pixel at a time.

[0072] There are two binary segmentations based on intensity (luma) and MV uniformities in the local sliding window, as illustrated in FIGS. 5A and 5B. Adjustment of a motion vector MV , which is either F_x and F_y , is described below. The operation described below for MV correction can be applied independently and in parallel for the two selected forward and backward MV F_x and F_y , received from MVS 204.

[0073] FIGS. 5A and 5B illustrate windows 520 and 522, respectively. Window 520 shown in FIG. 5A, where the cen-

tral pixel is not in a Halo region, is also illustrated in FIG. 4A. Window 522 shown in FIG. 5B, where the central pixel is in a Halo region, is illustrated in FIG. 4B.

[0074] According to the first binary segmentation, the window based on I_{11} is divided into two regions based on the luma level of each pixel relative to that of the central pixel in the window. All pixels with similar luma levels to the central pixel fall into one segment whereas pixels with different luma levels than that of the central pixel fall into the other segment.

The second binary segmentation in the window is based on the MV values for each pixel relative to that of the central pixel in the window. All pixels with similar MV values to that of the central pixel in the window fall into one segment, whereas pixels with different MV values than that of the central pixel fall into the other segment.

[0075] In combining these two binary segmentations, pixels in a window 520 about a central pixel fall into three groups of interest: group 1 502, group 2 504, and group 3 506. Group 1 502 consists of a local group of pixels with small weighted local error $WLRM = WLRM1 / WLRM3$ and similar luma levels and similar MV values to that of the central pixel as illustrated by FIG. 5A. Group 2 504 consists of a local group of pixels with large $WLRM$ and different luma level but similar MV values than that of the central pixel. Group 3 506 consists of a local group of pixels with small $WLRM$ and different luma levels and different MV values than that of the central pixel. Group 2 504 could be used as indication of a halo region. In some embodiments, only group 1 502 and group 2 504 are utilized. Group 1 502 can be utilized for MV smoothing. Group 2 504 can be utilized for Halo reduction.

[0076] In FIG. 5B, the central or considered pixel of window 522 is now in a suspected halo region, i.e. a pixel within group 2 504 of window 520. These two binary segmentations divide the window into three groups of interest: Group 1 508, group 2 510, and group 3 512. Group 1 508 consists of a local group of pixels with small $WLRM$ and similar MV value but different luma level to the central pixel of window 522. Group 2 510 consists of a local group of pixels with large $WLRM$ and similar luma levels and similar MV values to that of the central pixel of FIG. 5B. Group 3 512 consists of a local group of pixels with small $WLRM$ and different MV values but with similar luma levels to that of the central pixel of window 522. Group 2 510 could be used again as indication of a halo region. In some embodiments presented herein, only the last two groups, group 2 510 and group 3 512, are used for halo reduction. Once a halo region is confirmed, the corrected motion vector for the central pixel is based on motion vectors of group 3 512 for true halo reduction. For the smoothness purpose in MV field, the MV in each group of window 522 can be re-estimated using the local mean.

[0077] Although the dimensions of windows 520 and 522 can be of any size, in some embodiments, the dimensions of window 522 I by J may be large enough to sufficiently obtain pixels representing each of the groups when the current pixel is located in a suspected halo region. Moreover, I and J may be small enough to avoid wrong information which can falsify the MV estimation. In some embodiments, M12 202 utilizes a block-based approach with a block having $m \times n$ pixels, window 522 can have a (horizontal) length J chosen between n and $2.5n$ pixels and can have a window height I for economical purpose that is chosen between $0.5m$ to m .

[0078] In some embodiments, the MV in a given group of window 522 will be corrected if the MV estimation error

WFRM at the current pixel is larger than a threshold value and, in window 522 the number N_{DE} of pixels with small estimation error and different MV values to that of the central pixel is large enough. In this case, the corrected MV will be the mean of these different motion vectors that already yield small errors. Thus, as illustrated by FIG 4B, the MV in group 2 510 will be substituted by the mean value of motion vectors of group 3 512. In FIG 4A, the MV in group 1 502 will not be corrected.

[0079] If the above condition is not true, i.e. the MV estimation error is relatively big or N_{DE} is not important, the MV correction can be based on the cardinal numbers of the different groups. Let N_R be the number of pixels that resemble the central considered pixel for similar intensity and similar MV. Let N_F be the number of pixels of similar intensity but different MV values to that of the central pixel. Let N_{RD} be the number of pixels with similar intensities as the central pixel in the window. As a result, N_{RD} is equal to the sum of N_R and N_F . N_{RD} is smaller than the window size. As such, N_{RD} is a relative number whereas the window size $[*]$ is an absolute number indicating the size of the sliding window, windows 520 and 522.

[0080] FIG 6 illustrates MV selection or correction. As shown in FIG 6, in some embodiments MV selection or correction is a three-level decision in a two dimensional space based on the parameter N_R . The horizontal axis in FIG 6 is the absolute number, whose maximum is the window size $[*]$. The vertical axis is the relative number whose maximum is N_{RD} . If N_R is again the number of pixels of similar intensity and similar MV to that of the current central pixel, the selected MV can be one or none of two different estimated MV as shown in FIG 6. In the upper right corner where N_R is relatively equal to both cardinal numbers, the central pixel is assigned the mean of these same or similar motion vectors, which means that the mean of MV of group 1 502 in FIG 4A or FIG 5A. In the lower left corner, where N_R is relatively small in comparison with the two cardinal numbers, the central pixel is assigned the mean of these different motion vectors, which is the MV of group 3 512 in FIG 4B or FIG 5B. In the case when none of these criteria is met, no MV of the above groups is selected and the original MV produced in M11 210 is reused. Therefore, contrary to the prior art, the MV correction process utilized in some embodiments of the present invention do not require explicit detection of Covering/Uncovering regions.

[0081] In some embodiments, segmentation of the pixels in the local sliding window can be based on chroma levels. This gives additional confidence to the luma level segmentation described above with respect to FIGS 5A and 5B. The grouping of pixels in these embodiments may for example use luma and chroma difference on the one hand and motion vectors on the other hand.

[0082] It should be obvious to those skilled in the art that the segmentations of luma and chroma as well as motion vectors in a local sliding window as discussed above and illustrated in FIGS 4A, 4B, 5A and 5B are examples. In addition, the use of two cardinal numbers to partition the square illustrated in FIG 6 is also exemplary. One could envision alternative selection and other cardinal numbers without departing from the scope of disclosed embodiments.

[0083] The corrected motion vectors F_1 and B_1 , which correspond to the corrected F_0 and corrected B_0 , respectively, produced by Halo reduction 802, for example according to the above decision rules, are input to MV blending 804. In

order to avoid the spiky effect of any hard decision, the MV F_1 is blended with the selected MV F_0 of MVS 204 in a function of pixel-based forward estimation error (ef) of WFRM to provide MV output F_2 . F_2 is then given by

$$F_2 = F_0 + F_1 \cdot ef / (ef + 1) \quad (12)$$

In a similar manner, B_2 , the blended version of B_1 is given by

$$B_2 = B_0 + B_1 \cdot eb / (eb + 1) \quad (13)$$

where (eb) is the current pixel-based backward estimation error of WFRM.

[0084] The output from MV blending 804, F_2 and B_2 is input to Isolated MV Filtering 806. The obtained MV F_2 is then substituted by its P×Q neighbor median MV if the F_2 at the current pixel is relatively isolated. Precisely, the current MV output F_0 can be written as follows:

[0085] If in the window of size P×Q, the number of MVs similar to that of the current central pixel is smaller than a threshold value, then

$$F_0 = \text{median vector} \{ F_{(i,j)} | (i,j) \in \text{Window} \} \quad (14)$$

$$\text{if } (i,j) = F_{(0,0)} \text{ is the window central pixel index} \quad (15)$$

B_0 , the median filtered version of B_2 , can be obtained in similar fashion. The values F_0 and B_0 output from isolated MV filtering 806 is input to image 2nd interpolation 808.

[0086] In Image 2nd Interpolation 808, the above obtained motion vectors, F_0 and B_0 , are utilized in an interpolation. The 2nd interpolation is similar to the 1st interpolation M11 210 and may use one of the following interpolations:

$$I_2(x) = I_b + I_a(\alpha + \alpha F_0 + FF_0) + I_b(\alpha + \alpha B_0) \quad (16)$$

or

$$I_2(x) = I_b + I_a(\alpha + (1-\alpha)F_0) + I_b(\alpha + (1-\alpha)B_0) \quad (17)$$

$$I_2(x) = I_b + I_a(\alpha + \alpha F_0) + I_b(\alpha + \alpha B_0) \quad (18)$$

$$I_2(x) = I_b + I_a(\alpha + \alpha F_0) + I_b(\alpha + (1-\alpha)B_0) \quad (19)$$

When small motion jitter is tolerable in a multi-frame interpolation, the following interpolation can be considered instead of the interpolation of Eqs. 16 or 17. This interpolation may increase the picture sharpness and result in more halo reduction. In this interpolation, the alpha value α is modified non-linearly by α' as follows (quadratic modification for example):

[0087] If $0 \leq \alpha \leq 1/2$ then $\alpha' = 2\alpha^2$. If $1/2 < \alpha < 1$ then $\alpha' = 1 - 2(1 - \alpha)^2$.

$$I_2(x) = I_b + I_a(\alpha' + (1-\alpha')F_0) + I_b(\alpha' + (1-\alpha')B_0) \quad (20)$$

$$I_2(x) = I_b + I_a(\alpha' + \alpha' F_0) + I_b(\alpha' + \alpha' B_0) \quad (21)$$

$$I_2(x) = I_b + I_a(\alpha' + \alpha' F_0) + I_b(\alpha' + (1-\alpha')B_0) \quad (22)$$

[0088] The output from M12 206 as shown in FIG 2 is the interpolated alpha-plane image 102, I_2 , along with forward and backward motion vectors F_1 and B_1 . Post Processor (PP) 208 receives the image data I_2 and the motion vectors F_1 and B_1 , and may reduce again some remaining halo in the presence of a lattice background and may also reduce flickering effects caused by isolated erroneous interpolated pixels in the occlusion regions or resulting from isolated MVs.

US 2009/0161010 A1

Jun 25, 2009

7

[0089] FIG 9 illustrates a Halo reducer 900, which is part of post processor 208, for use with a lattice in the background. When a lattice is present in the background, small MV errors in occlusion regions can break the lattice structure in the interpolated image and the effect will be objectionable. As shown in FIG 9, halo reducer 900 can include a Detector of Lattice Background in Occlusion Regions 902, a Lattice MV Corrector 904, an Image Interpolator for Lattice background 908, and a Soft Mixer 910.

[0090] The Occlusion Regions Detection in detector 902 can be based on whether or not SADF or SADB is greater than a threshold value. Moreover, in the occlusion regions, the forward and backward motion vectors are generally not aligned, i.e. the absolute of the sum (F_n+B_n) is larger than a threshold value. It is worthwhile to note that detector 902 detects occlusion regions, but does not need to detect separate covering or uncovering areas. The lattice detection can be accomplished by block-based FFI analysis in MF 202. The binary signal PI RD represents the periodic detection on/off result of a lattice. Furthermore, in order to correct the erroneous MV, a component-wise comparison can be performed between the vectors F_n and G and B_n and G , where the MV G is defined as a global MV estimated via suitable histogram analysis in ME 202. For example if the background is still or MV G is not available, G can be set equal to 0. The MV G can be also estimated by using a suitable sliding window locally segmented to three pixel groups as discussed above with respect to M12 206. The detector binary signal output DI BOR activates the Lattice MV Corrector 904. DI BOR is applied also to the input of a Low Pass Filter 906 which yields in turn an analogue signal (a) which is input to soft mixer 910.

[0091] Lattice MV Corrector 904 when activated by detector signal DI BOR substitutes F_n and B_n by G to produce motion vectors F_c and B_c . Accordingly, F_c and B_c are the forward and backward motion vectors provided by MV corrector.

[0092] Image Interpolator for Lattice background 908 when the motion vectors F_c and B_c are available can be described by the previous interpolation model of LQ 16. The Image Interpolation result, I_2 , can, then be defined as follows:

$$I_2(x) = B I_1(x+d) + (1-I_1(x)) I_1(x) \quad (19)$$

[0093] In order to reduce the detection boundary effect, soft mixer 910 is utilized to provide an image output I_M for the halo-lattice correction.

$$I_M(x) = I_1(x) + I_2(x) - I_1(x) I_2(x) \quad (20)$$

The image output I_M can then be applied to a Flicker Reducer for the final correction.

[0094] FIG 10 illustrates a Flicker Reducer 1000, which is part of post processor 208. Flicker reducer 1000 can be the final stage of post processor 208 shown in FIG 2. As shown in FIG 10, flicker reducer 1000 can include a flicker detector 1002, a neighbor average calculator 1004, and a soft mixer 1008.

[0095] Flicker detector 1002 detects possible flickering effect which is due primarily to isolated erroneous interpolated pixels in occlusion regions or to isolated MVs. Occlusion regions detection previously described is based on SADF SADB the sum of forward and backward motion vectors, and the difference between the current intensity and its neighbor (for example eight-neighbor) average value. The isolated MV detection is based on, in a small window, the number of pixels with the different MV from the current one. The difference is considered for each horizontal or vertical

component of forward and backward MV F_n and B_n . The binary signal output of the detector is applied to a Low Pass Filter 1006, which yields in turn an analogue signal (b) for further mixing purposes. Average Calculator 1004 provides at its output an intensity average value I_M of neighboring pixels (for example eight neighboring pixels) to the current pixel.

[0096] Soft Mixer 1008 provides the final image I_f according to the following relation:

$$I_f(x) = I_M(x) + (I_1(x) - I_M(x))k^k \quad (21)$$

The image $I_f(x)$, then represents the image in the alpha plane 102 shown in FIG 1.

[0097] In some embodiments, interpolating more than one frame between existing frames is performed. For example converting 24 fps film content to 120 Hz video requires interpolating four frames between any two original frames, as is illustrated by alpha-planes 702, 704, 706, and 708 in FIG 7. Motion estimation (ME) in ME 202 is performed between the original two images 104 and 106 I_n and I_{n+1} , respectively, and it is therefore the same as the method described above. The MVT and MVB selection in this case is the same as the example of the single frame interpolation given above. Accordingly, motion vector selection (MVS) 204 performs similarly to that described above. For each interpolated plane, a similar MVS can be used in a suitable combination of parallel and/or series as a function of available calculation power. The first motion interpolation M11 210 can be accomplished similarly by estimating pixel values at different time intersection of the motion vectors. The second motion interpolation performed in M12 is also similar for each alpha plane. Finally, for high frame rate output, MVS 204, M11 210, M12 206, and PP 208 can be performed in a suitable combination of parallel and/or series processing as a function of available calculation power.

[0098] Embodiments of the application can be implemented on any machine capable of processing image data. For example, embodiments can be implemented on a processor executing software code that implements the functions discussed here. Such software can be stored on any computer medium, including, for example, hard drives, memory removable drives. Additionally, embodiments can be implemented on a dedicated ASIC with a processor and memory.

[0099] Some embodiments of the present invention can significantly reduce the appearance of halo flickering and block artifact in interpolated images for real-time applications. It should be obvious to those skilled in the art that this selection of MV and the corresponding interpolation of the intermediate α -plane are exemplary and other multi-level decisions could be made or different groups may be considered differently to select/correct MV and generate an intermediate image without departing from the scope of the embodiments given herein. It should be appreciated by those skilled in the art that certain steps or components could be altered without departing from the scope of the embodiments presented herein. As such, the present invention should be limited only by the following claims.

We claim,

1. A method for interpolating an image between first and second adjacent images comprising:
 - estimating a block-based forward motion vector and a block-based backward motion vector between the first and second adjacent images;

- selecting a forward motion vector for each pixel in the image based on the block-based forward motion vector and a forward error function
- selecting a backward motion vector for each pixel in the image based on the block-based backward motion vector and a backward error function,
- interpolating a first image from the selected backward motion vectors, the selected forward motion vectors, a block-based forward estimation error and a block-based backward estimation error
- interpolating a second image from the first image based on forward and backward weighted errors of the selected forward motion vector and the selected backward motion vector and
- determining the image from the second image
- 2 The method of claim 1, wherein estimating the block-based forward estimator and estimating the block-based backward estimator includes calculating a forward sum of absolute differences error and a backward sum of absolute differences error
- 3 The method of claim 1 wherein estimating the block-based forward motion vector and the block-based backward motion vector is adaptive.
- 4 The method of claim 1, wherein estimating the block-based forward motion vector and the block-based backward motion vector is performed on interlaced images
- 5 The method of claim 1 wherein selecting the forward motion vector or selecting the backward motion vector includes converting block-based motion vectors to pixel-based motion vectors of the image
- 6 The method of claim 1, wherein the forward error function and the backward error function is the sum of the differences between motion vectors in the image estimated from the first and second adjacent images.
- 7 The method of claim 1 wherein the forward error function and the backward error function is a composite local error
- 8 The method of claim 1, wherein the forward error function and the backward error function is a weighted local error
- 9 The method of claim 1, wherein interpolating the first image includes
- calculating a number of pixels based on pixel based values of block error functions and a location α of the image relative to the first and second adjacent planes
 - calculating the first image based on the selected forward motion vector, the selected backward motion vector a normalized number of pixels based on the number of pixels, and the location α .
- 10 The method of claim 9 wherein interpolating the first image is a context based interpolation
- 11 The method of claim 9 wherein interpolating the first image includes a function of temporal distance
- 12 The method of claim 1 wherein interpolating the second image includes
- reducing a Halo effect,
 - blending motion vectors,
 - filtering motion vectors, and
 - image interpolating
- 13 The method of claim 12 wherein reducing the halo effect includes
- segmenting from local intensity information in a plane of the image, the selected forward motion vector, and a corresponding local weighted forward minimum error a local sliding segmented window having a plurality of pixel groups of interest for forward motion vectors
 - segmenting from the local intensity information the plane of the image, the selected backward motion vector, and a corresponding local weighted backward minimum error a local sliding segmented window having a plurality of pixel groups of interest for backward motion vectors,
 - correcting the selected forward motion vector based on the current position of one of the plurality of pixel group of interest for providing a corrected forward motion vector that reduces a halo effect, and
 - correcting the selected backward motion vector based on the current position of one of the plurality of pixel group of interest for providing a corrected backward motion vector that reduces a halo effect
- 14 The method of claim 13 wherein blending motion vectors includes calculating an adjusted forward motion vector by adding a forward estimated error times a difference between the corrected forward motion vector and the selected forward motion vector to the selected forward motion vector
- 15 The method of claim 14, wherein blending motion vectors further includes calculating an adjusted backward motion vector by adding a backward estimated error times a difference between the corrected backward motion vector and the selected backward motion vector to the selected backward motion vector
- 16 The method of claim 15, wherein filtering motion vectors includes substituting the adjusted forward motion vector with a median forward motion vector and the adjusted backward motion vector with a median backward motion vector
- 17 The method of claim 16, wherein image interpolating includes forming the second image from the adjusted forward motion vector and the adjusted backward motion vector
- 18 The method of claim 17 wherein forming the second image includes interpolating with a quadratic modification of the temporal location of the image
- 19 The method of claim 1 wherein determining the image includes post processing the second image by
- reducing a halo effect and
 - reducing a flicker effect
- 20 The method of claim 19, wherein reducing the halo effect includes
- detecting a lattice background,
 - correcting motion vectors
 - interpolating a final image, and
 - mixing a correction into the final image
- 21 The method of claim 20 wherein reducing the flicker effect includes
- detecting flicker,
 - determining neighborhood average and
 - mixing the neighborhood average with the final image to form the image.
- 22 An image interpolator, comprising
- a block-based motion estimator coupled to receive adjacent images, the block-based motion estimator providing a block-based motion vector, a block-based motion vector, a forward error, and a backward error,
 - a motion vector select coupled to the block-based motion estimator the motion vector select providing pixel based selected forward motion vector and selected backward motion vector, and providing pixel based forward and backward errors,

US 2009/0161010 A1

Jun 25, 2009

9

- a first image interpolator coupled to the block-based motion estimator and the motion vector select, the first image interpolator providing a first interpolated image from the selected forward motion vector and the selected backward motion vector,
 - a second image interpolator coupled to the first image interpolator and the motion vector select, the second image interpolator providing a second interpolated image and corrected forward motion vectors and corrected backward motion vectors, and
 - a post processing block coupled to the first image interpolator and the second image interpolator, the post processing block providing a final interpolated image from the second interpolated image and the corrected forward motion vectors and the corrected backward motion vectors
- 23** The image interpolator of claim 22, wherein the second image interpolator includes
- a halo reducer that utilizes a sliding window to correct the selected forward motion vector and the selected backward motion vector,
 - a MV blending block that further corrects the selected forward motion vector and the selected backward motion vector
 - an isolated MV filter that further corrects the selected forward motion vector and the selected backward motion vector to produce the corrected forward motion vector and the corrected backward motion vector, and
 - a second image interpolator that interpolates an image from the corrected forward motion vector and the corrected backward motion vector
- 24** The image interpolator of claim 22, wherein the post processing block includes
- a halo reducer, and
 - a flicker reducer

- 25** The image interpolator of claim 24, wherein the halo reducer includes
- a lattice background detector
 - a lattice motion vector corrector coupled to the latter background detector, the lattice motion vector corrector correcting the corrected forward motion vector and the corrected backward motion vector, and
 - an image interpolator that interpolates an image from motion vectors corrected by the lattice motion vector corrector
- 26** The image interpolator of claim 24, wherein the flicker reducer includes
- a isolated element detector, the isolated element detector detecting an isolated motion vector or an isolated pixel and
 - a neighbor average calculator
- 27** An image interpolator, comprising
- a motion estimator that estimates block-based motion vectors between a first image and a second image
 - a motion vector select coupled to the motion estimator that provides selected motion vectors based on the block-based motion vectors,
 - a first interpolator coupled to receive the selected motion vectors and provide a first interpolated image for a plane between the first image and the second image, and
 - a second interpolator coupled to receive the first interpolated image that corrects the selected motion vectors to form corrected motion vectors and provides a second interpolated image based on corrected motion vectors

* * * * *

LISTE DES RÉFÉRENCES

- Aggarwal, J K et Nandhakumar, N (août 1988) On the computation of motion from sequences of images – a review *Proceedings of the IEEE*, volume 76, numéro 8, p 917-935
- Astola J , Haavisto, P et Neuvo, Y (avril 1990) Vector Median Filters *Proceedings of the IEEE*, volume 78, numéro 4, p 678–689
- Bellers, E B , Van Gurp, J W , Janssen, J G W M , Braspenning, J R et Wittebrood, R (janvier 2007) Solving occlusion in Frame-Rate up-Conversion *IEEE International Conference on Consumer Electronics*, p 1-2, Digital Object Identifier 10 1109/ICCE 2007 341494
- Beon, N , Min, J et Jung, B (mars 2008) *High-Definition Image Display Device and Method of Converting Frame Rate Thereof* U S Patent Application Publication number 2008/0074350A1
- Bierling, M (janvier 1988) Displacement Estimation by Hierarchical Block-Matching *Proceedings of SPIE Conference On Visual Communication And Image Processing*, volume 1001, p 942-951
- Blanchfield, P , Wang, D , Vincent, A , Speranza, F et Renaud, R (2006) Advanced Frame Rate Conversion and Performance Evaluation *SMPTE Motion Imaging Journal*, volume 115, numéro 4, p 153-159
- Blume, H , Herczeg, G , Erdler, O et Noll, T G (août 2002) Object based refinement of motion vector field applying probabilistic homogenization rules *IEEE Transactions on Consumer Electronics*, volume 48, numéro 3, p 694–701
- Burl, M , Thomson, R et Layton, P (avril 2000) Video Image Motion Compensation using an Algorithm Involving at least Two Fields *British Broadcasting, London, United Kingdom* U S Patent number 6011596
- Burt, P J et Adelson, E H (avril 1983) The laplacian pyramid as a compact image code *IEEE Transactions on Communications*, volume 31, numéro 4, p 532-540
- Castagno, R , Haavisto, P et Ramponi, G (octobre 1996) A method for motion adaptive frame rate up- conversion *IEEE Transactions on Circuits and Systems for Video Technology*, volume 6, numéro 5, p 436-445
- Chan, M H , Yu, Y B et Constantinides, A G (août 1990) Variable size block matching motion compensation with application to video coding *IEE Proceedings on Communications, Speech and Vision* volume 137, numéro 4, p 205–212

-
- Chang, C , et Yanagihara, N (octobre 2000) *Hierarchical Motion Estimation Process and System using Block-Matching and Integral Projection* U S Patent number 6130912
- Chen, L , Lee, Y et Lai, Y (décembre 2000) Motion Estimation Employing a Three-Step Hierarchical Search Block-Matching Algorithm U S Patent Number 6160850
- Cheng, Y (août 1995) Mean Shift, Mode Seeking, and Clustering *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 17, numéro 8, p 790-799
- Choi, B D , Han, J W , Kim, C S et Ko, S J (août 2006) Frame Rate Up-Conversion Using Perspective Transform *IEEE Transactions on Consumer Electronics*, volume 52, numéro 3, p 975-982
- Choi, B D , Han, J W , Kim, C S et Ko, S J (avril 2007) Motion-Compensated Frame Interpolation Using Bilateral Motion Estimation and Adaptive Overlapped Block Motion Compensation *IEEE Transactions on Circuits and Systems for Video Technology*, volume 17, numéro 4, p 407-759
- Choi, B D , Han, J W , Kim, C S , et Ko, S J (août 2006) Frame Rate Up-Conversion Using Perspective Transform”, *IEEE Transactions on Consumer Electronics*, volume 52, numéro 3, p 975-982
- Choi, B T , Lee, S H et Ko, S J (août 2000) New Frame Rate Up-Conversion using Bi-Directional Motion Estimation *IEEE Transactions on Consumer Electronics*, volume 46, numéro 3, p 603-609
- Dane, G et Nguyen, T Q (2004) Smooth motion vector resampling for standard compatible video post-processing *38th Asilomar Conference on Signals, Systems and Computers*
- De Haan, G (2000) Chapter 4 Picture rate conversion, *Video Processing for Multimedia Systems* University Press, Eindhoven ISBN 90-9014015-8, p 115-140
- De Haan, G et Mertens, M (mars 2006) *Motion estimator for reduced halos in MC up-conversion* U S Patent number 7010039
- De Haan, G et Pelagotti, A (avril 2001) *Motion Vector Estimation and Detection of Covered/Uncovered Image Parts*, U S Patent number 6219436
- De Haan, G et Pelagotti, A (juin 2006) *Problem Area Location in an Image Signal* U S Patent 7058227 B2
- De Haan, G et Pelagotti, A (novembre 2002) *Problem Area Location in an Image Signal* U S Patent 6487313
- De Haan, G P , Biezen, W A C , Huijgen, H et Ojo, O A (octobre 1993) True-motion estimation with 3-D recursive search block matching *IEEE Transactions on Circuits and Systems For Video Technology*, volume 3, numéro 5, p 368-379

-
- De Haan, G et Mertens, M J W (mars 2006) *Motion estimator for reduced halos in MC up-conversion* U S Patent number 7010039
- Fujiwara, S , et Taguchi, A (décembre 2005) Motion-compensated frame rate up-conversion based on block matching algorithm with multi-size blocks *Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems*
- Fukunaga, K et Hostetler, L (1975) The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition," *IEEE Transactions on Information Theory*, volume 21, numéro 1, p 32-40
- Ha, T , Lee, S et Kim, J (mai 2004) Motion Compensated Frame interpolation by new Block-based Motion Estimation Algorithm *IEEE Transactions on Consumer Electronics*, volume 50, numéro 2, p 752-759-1838
- Hilman, K , Park, H W , et Kim, Y (septembre 2000) Using motion compensated frame-rate conversion for the correction of 3 2 pulldown artifacts in video sequences *IEEE Transactions on Circuits and Systems for Video Technology*, volume 10, numéro 6, p 869-877
- Hong, S , Park, J H et Berkeley, B H (2006) Motion-Interpolated FRC Algorithm for 120 Hz LCD *SID International Symposium Digest of Technical Papers*, volume 37, numéro 2, p 1892-1895
- Huang, A M et Nguyen, T Q (mai 2008) A Multistage Motion Vector Processing Methods for Motion-Compensated Frame Interpolation *IEEE Transactions on Image Processing*, volume 17, numéro 5, p 694-708
- Huang, A M et Nguyen, T Q (avril 2009) Correlation-Based Motion Vector Processing With Adaptive Interpolation Scheme for Motion-Compensated Frame Interpolation *IEEE Transactions on Image Processing*, volume 18, numéro 4, p 740-752
- Jeon, B W , Lee, G I, Lee, S H et Park, R H (août 2003) Coarse to Fine frame interpolation for frame rate up-conversion using pyramidal structure *IEEE Transactions on Consumer Electronics*, volume 49, numéro 3, p 499-508
- Jung, H (mai 1998) *Method for Determining Motion Vectors using a Hierarchical Motion Estimation* U S Patent number 5754237
- Kang, S J , Cho, K R et Kim, Y H (novembre 2007) Motion Compensated Frame Rate Up-Conversion Using Extended Bilateral Motion Estimation *IEEE Transactions on Consumer Electronics*, volume 53, numéro 4, p 1759-1767
- Kang, S , Yoo, D , Lee, S et Kim, Y H (novembre 2008) Multiframe-based bilateral motion estimation with emphasis on stationary caption processing for frame rate up-conversion *IEEE Transactions on Consumer Electronics*, volume 54, numéro 4, p 1830-1838

-
- Kaup, A et Aach, T (avril 1994) Efficient prediction of uncovered background in interframe coding using spatial extrapolation *Proceedings of the 1994 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, p 501-504, Adelaide, Australia
- Krishnamurthy, R , Woods, J W et Moulin, P (août 1999) Frame interpolation and bidirectional prediction of video using compactly encoded optical-flow fields and label fields *IEEE Transactions on Circuits and Systems for Video Technology*, volume 9, numéro 5, p 713-726
- Kuo, Y et Kou, C -C J (juillet 1998) Motion-compensated interpolation for low-bit-rate video quality enhancement *Proceedings of SPIE Conference on Applications of Digital Image Processing XXI* , volume 3460, p 277-288
- LeDinh, C T (2003) *Traitement Vidéo, Notes de cours GEI 754*
- LeDinh, C T (2008) *Ingénierie des systèmes numériques Notes de cours GEI 759 [LD08]*
- LeDinh, C T (juillet 2006) Apparatus and Methods for Adaptive Spatial Segmentation-Based Noise Reducing for Encoded Image Signal *US Patent number 7076113*
- Lee, G , Jeon, B -W , Park, R -H et Lee, S -H (juin 2003) Hierarchical Motion Compensated Frame Rate Up-Conversion Based on the Gaussian/Laplacian Pyramid *IEEE International Conference on Consumer Electronics*, p 350-351
- Lee, J , Jang, S , Chen, O S , Kim, Y et Kim, C (novembre 2007) An Efficient Frame Rate Up-Conversion Methode for Mobile Phone with Projection Functionality *IEEE Transactions on Consumer Electronics*, volume 53, numéro 4, p 1615-1621
- Lee, S H , Kwon, O et Park, R H (août 2003) Weighted-Adaptive Motion-Compensated Frame Rate Up-Conversion *IEEE Transactions on Consumer Electronics*, volume 49, numéro 3, p 485-492
- Lee, S -H , Shin, Y -C , Yang, S , Moon, H -H et Park, R -H (août 2002) Adaptive motion-compensated interpolation for frame rate up-conversion *IEEE Transactions on Consumer Electronics*, volume 48, numéro 3, p 444-450
- Li, W et Salari, E (janvier 1995) Successive Elimination Algorithm for Motion Estimation *IEEE Transactions on Image Processing*, volume 4, p 105-107
- Li, Y (mai 2007) *Motion Compensated Frame Interpolation Apparatus and Method* U S Patent Application Publication numéro 2007/0121725A1
- Ling, Y , Wang, J , Liu, Y et Zhang, W (mai 2008) A novel spatial and temporal correlation integrated based motion-compensated interpolation for frame rate up-conversion *Transactions on Consumer Electronics*, volume 54, numéro2, p 863-869

-
- Liu, B et Zaccarin, A (avril 1993) New Fast Algorithms for the Estimation of Block Motion Vectors *IEEE Transactions on Circuits and Systems for Video Technology*, volume 3, numéro 2, p 148-157
- Rossignol, F , Tran, T H , et LeDinh, C T (avril 2010) *System and Method for Motion Vector Filtering based on Local Image Segmentation and Lattice Maps*, Patent Application Serial No 12/719,785, avril 2010
- Pelagotti, A , Schutten, R J (mai 2006) *Reducing Halo-Like Effects in Motion-Compensated Interpolation* U S Patent number 7039109
- Rossignol, F , LeDinh, C T et Tran, T H (juin 2009) *Motion Estimation with an Adaptive Search Range* U S Patent Application Numéro 2009/0161763 A1
- Schutten, R J , et De Haan, G (août 1994) Real-time 2-3 pull-down elimination applying motion estimation/compensation in a programmable device *IEEE Transactions on Consumer Electronics*, volume 4, numero 3, p 930-938
- Shi, F et Raveendran, V R (décembre 2005) *Method and Apparatus for Motion Compensated Frame Rate Up Conversion for Block-Based Low Bit Rate Video* U S Patent Application Publication No 2005/0265451 A1
- Sugiyama, K , Aoki, T et Hangai, S (novembre 2005) Motion Compensated Frame Rate Conversion using Normarized Motion Estimation *IEEE Workshop on Signal Processing Systems Design and Implementation*, p 663-668
- Tekalp, A M (1995) *Digital Video Processing* Prentice Hall inc , Upper Saddle River, New Jersey, p 81-92
- Thoma, R et Bierling, M (1989) Motion compensating interpolation considering covered and uncovered background *Signal Processing Image Communication*, volume 1, p 191-212
- Thomas, G A , Burl, M (décembre 1999) *Vector Assignment for Video Image Motion Compensation* BBC, Broadcasting House, London, United Kingdom U S Patent number 6005639
- Thomas, G A , Burl, M (décembre 1999) *Vector Assignment for Video Image Motion Compensation* BBC, Broadcasting House, London, United Kingdom U S Patent number 6005639
- Tran, T H et LeDinh, C T (avril 2011) Frame Rate Converter with Pixel-based Motion Vectors Selection and Halo Reduction using Preliminary Interpolation *IEEE Journal of Selected Topics in Signal Processing* Special Issue on Recent Advances in Video Processing for Consumer Displays RAVPCD
- Tran, T H et LeDinh, C -T (juin 2009) *Image Interpolation with Halo Reduction* U S Patent Application number US 2009/0161010A1

-
- Tran, T H , LeDinh, P T , Rossignol, F et LeDinh, C T (avril 2010) High-definition Frame Rate Converter using a new hierarchical approach with halo reduction consideration Patent Application Serial No 12/761,214, April 15 2010
- Uchida, M , Kondo, T , Nakaya, H , Horishi, T et Ishizaka, T (mars 1997) *Motion Vector Detection using Hierarchical Calculation* U S Patent number 5610658
- Vatolin, D et Grishin, S (juillet 2006) N-times Video Frame-rate Up-conversion Algorithm based on Pixel Motion Compensation with Occlusions Processing "*Graphucon*", *International Conference on Computer Graphics & Vision*, p 112-119
- Wang, D et Vincent, A (septembre 2003) *Method for Temporal Interpolation of an Image Sequence using Object-Based Image Analysis* U S Patent number 6625333 B1
- Wang, Y , Ostermann, J et Zhang Y Q (2002) *Video Processing and Communications*, Prentice-Hall Inc
- Wang, Z , Bovik, A C , Sheikh, H R et Simoncelli, E P (avril 2004) Image Quality Assessment From Error Visibility to Structural Similarity *IEEE Transactions on Image Processing*, volume 13, numéro 4, p 600-612